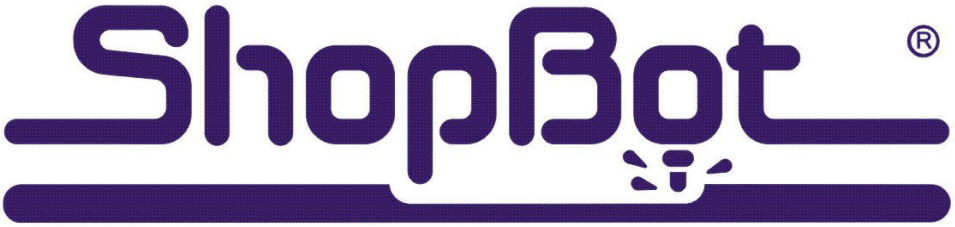


ShopBot User Guide

This manual is part of a suite of documents for your ShopBot that includes: the *Users Guide* on using your ShopBot and the ShopBot Control Software; the *Command Reference* documenting the specific format and usage for every ShopBot Command; and the *Part Programming Handbook* detailing the functions available in the ShopBot Part File Language and shows how these functions can be used in your own projects.

This manual is specifically intended for PRS ShopBots and Buddies (or PRSalpha or PRSstandards with Version 4g Control Boards). Manuals for some earlier ShopBots may be included with the software downloads for those tools from the ShopBot Tools website.

ShopBot Tools, Inc.
3333-B Industrial Drive
Durham, NC 27704
919-680-4800 (phone)
919-680-4900 (fax)
info@ShopBotTools.com
http://www.ShopBotTools.com

ShopBot is a Registered Trademark of ShopBot Tools, Inc. Other brand and product names are trademarks or registered trademarks of their respective holders.
U.S. Patent #6,070,106 for the ShopBot PRS Tool.
Manual Illustrations by Don Elliott.

Copyright (c) 1996-2008 by ShopBot Tools, Inc.
All Rights Reserved.
Related software Copyright (c)1996-2008 by W.G. Hall and ShopBot Tools, Inc.
All Rights Reserved.

Table of Contents

Welcome to ShopBot! 4
You Need to Know This: Warnings 5
Getting Started with Your ShopBot Control Software 7
- The ShopBot Control Software 7
- Installing the Software 7
- Starting Up 11
- The Basic ShopBot Command Principle 12
- Entering Values (Parameters) 12
- So Let's Run a Part File 14
- Getting Help 16

The Layout of Your ShopBot Table: Directions, Distances, and Base Coordinates 17
- The Logic of XYZ 18
- Giving Directions; ABSOLUTE vs RELATIVE distance 18
- You Can Reorganize Your "Graph Paper" 19
- Temporary Zero Points When Cutting Files 19
- Table Base Coordinates vs Current Working Coordinates 20
- Software Limit Checking 21
- Maintaining Accurate XYZ Locations 21
- Changing the Direction of Movement of an Axis 21
- Software Axis Designation 22
- Optional 4th or 5th Axis or Channel 22
- Inches or Millimeters 22

What Your Tool Can Do 23
- Giving Action Commands 24
- Changing Settings and Values (Commands) 24
- Utility Commands 25

ShopBot Part Files (.sbp) 25
- What's in a Part File 25
- Using the Editor 26
- Automatic Recording of Things You Do at the Keyboard 27
- Don't Let this Scare You... 27
- Creating Part Files from other Software 30

From Your Ideas to a ShopBot Part File 32
- Representation to Realization 32

A First Project: Surfacing Your Table 36
Working with Your Computer 38
Working with Your Tool 39
- Speeds, Feed-Rates and Rapid-Positioning 39
- Acceleration and Deceleration 'Ramping' 42
- Direction of Cut and Tool Path 44
- Tooling 44
- Clamping Your Material 45
- Dust Collection 46

Working with Accessories 47
- Input Switches, Limit Switches and STOP Inputs 47
- ShopBot's Remote 'Stop' Button 48
- Output Switches 48
- Additional Axes, Rotary Indexers, Lathes 49
- 3-D Digitizing (Probing Surfaces) 49

Working with G-code 50
Working with the ShopBot Converters for .dxf, .hpg/.plt, .bmp, and .jpg 58
- Starting a Conversion 59
- Specific Info for Each Type of Conversion 59
- Conversion from HPGL Plotter File Format: 61

Maintenance 64
- Mechanical Adjustment - X & Y Axes 64
- Electrical/Electronic 65
- Control Software 66

Troubleshooting 66
- First get your thinking right … 67

Copyright 2008 ShopBot Tools, Inc

Dealing with Tool Operation Problems: Primary Failures 68
Intermittent Connection Problem: 68
Problems with Movement 68
Changing the direction of a motor....69
Determining whether you have a software, electrical, or mechanical problem when an axis moves but does not go the correct distance....69
Dealing with Tool Operation Problems: Qualitative Issues 71
Getting Help....72
ShopBot FAX FORM 74
Technical Information....75
Specifications 75
ShopBot Computer Requirements....76
ShopBot Philosophy 77

 Copyright 2008 ShopBot Tools, Inc

Welcome to ShopBot!

Your ShopBot router is a low-cost, computer-numerically-controlled router (CNC Router). It is a workhorse of a workshop robot -- a Personal Robotic Tool. A ShopBot can perform precision, large-format, cutting, drilling, machining, and shaping in many materials and is controlled by a standard personal computer. A ShopBot offers you new ways to be productive and new paths to creativity. With your ShopBot you can mass produce a single item, or explore variations of plans, patterns, shapes and forms, and their reproduction in ways that would be difficult to attempt manually.

A ShopBot is a tool you can work with in amazing ways. And just as important, you can understand it, afford it, and maintain it -- it is the kind of technology you can live with.

Copyright 2008 ShopBot Tools, Inc

You Need to Know This: Warnings

ALERT! HOW TO STOP!!

During tool operation the SPACEBAR on your computer keyboard becomes a Panic/Stop/Halt button. Hitting it will immediately stop the tool's movement.

Your ShopBot is also supplied with a **STOP Button** that you can place in a convenient location on your tool. Hitting this button will stop the tool's movement. On a PRSalpha ShopBot or Buddy the STOP Button will also cut power to the spindle/router.

TOOLS REQUIRE ATTENTION! Follow Safe Procedures!

Your ShopBot, in conjunction with a router or other power tool, is a flexible, tool-movement system that can reduce your woodworking risks by providing a method of cutting wood or other material without having to interact with the cutting device or the material during the cutting process. However, as with all power tools, care and attention are required to use a ShopBot safely. ShopBot Tools, Inc. assumes you will use this product safely and follow accepted safety precautions for woodworking and machining.

READ THESE SAFETY INSTRUCTIONS

1. **READ**. Read this manual well to acquaint yourself with how to handle the tool safely and effectively before use. Read related manuals for the router and/or spindle that will be mounted on your ShopBot.

2. **PRACTICE**. Practice operating your ShopBot tool with your computer and the ShopBot Control Software BEFORE activating the router or spindle.

3. **PROTECT**. Turn OFF your router or other power tool before loading or positioning a workpiece or adjusting the position of the tool. Do not change router bits or other cutters without first unplugging the power tool or having a positive system to make sure the power tool is not accidentally activated. It is your responsibility to use it safely.

4. **HOLD-DOWN**. Never attempt to cut wood or other material without first ensuring that the work is firmly secured to the work surface. Power cutting tools always carry the risk that work material or broken cutters will be sent flying towards the operator or others in the area. A tool like ShopBot allows you to work with the workpiece secured and thus greatly reduces this risk. It also allows the operator to be away from the cutting and protected. However, YOU must take the steps to secure the workpiece and protect the operator. Note that not only must the full starting piece be secured, but you need to attend to the attachment and security of any pieces that will become cutouts or cutoffs during the cutting or machining process. Stand clear of the tool when it is in operation and protected from debris, parts, or broken cutters that might fly out during operation. Never attempt to push material by hand through a moving bit or to interact with the tool while it is running.

5. **AWAKE**. Never operate the tool when you are fatigued.

6. **EYES & EARS**. Always protect your eyes and ears when operating your ShopBot.

7. **ATTEND**. NEVER leave the tool unattended while it is running. A cutting error or workpiece slippage that the tool cannot detect might occur. Some person unknowledgeable

Copyright 2008 ShopBot Tools, Inc

about tool operation might approach and start the tool. Or some other unexpected event might occur.

8. **GUARD**. Always position the dust skirt correctly to guard against flying particles.

9. **TO STOP**. During a cutting or motion process, the SPACEBAR on the computer keyboard is a Panic/Stop/Halt button. Hitting the bar will stop the tool's movement. Your ShopBot PRSalpha also has a remote STOP Button that interrupts power to the stepper motors and the spindle or router after it is pushed. You should locate the STOP Button convenient to your tool and workstation. **Note: For PRS systems the power tool is operated independently of your ShopBot, it must be turned off separately.**

10. **POSITION**. The safest location for you during the operation of the tool is within easy reach of the computer keyboard or STOP Button and well away from the path of the tool. Because bits can break and fly loose during cutting -- stand behind a protective screen.

11. **BE SMART**. Most importantly, never place yourself at risk during a cutting or machining process by placing any part of you near the cutting path or by attempting to move or adjust the workpiece or active tool. SHOPBOT IS A ROBOT, BUT YOU ACTIVATE ITS MOVEMENT AND YOU TURN ON AND OFF ITS POWERED CUTTING-TOOL.

This Manual Assumes …

This manual has been written with the assumption that you are familiar with basic aspects of woodworking techniques for safe operation of the power tools and the basic operation of your computer. Information in this manual is subject to change without notice.

We've done our best to make ShopBot a safe and capable tool. We will work to make you happy with your ShopBot, but we need to make each of our responsibilities clear.

ShopBot assumes no responsibility for any damage to property or person resulting from use of power tools with our products. Safe operation of our tools and appropriate workshop precautions will place personnel in locations where they should not be subject to injury.

ShopBot Tools has provided protection and isolation circuitry in the controller/driver board that links your computer to the tool. We assume no responsibility for damage that may occur to your computer as a result of power surges, static discharges, or unexpected electrical events occurring in the system. We further assume no responsibility for damage to your computer that may occur because it is being operated in a workshop environment.

Your ShopBot CNC Router is both a physical tool and computer software. We strongly recommend that you keep separate permanent records of all important computer data, files, and programs. Data may be lost or altered with virtually any software product under certain circumstances. Therefore, ShopBot assumes no responsibility for data lost or otherwise rendered unusable whether as a result of improper use, repairs or modification, defect, or any other cause.

In no event will ShopBot assume responsibility for damages arising out of the use or inability to use this product (including without limitation damages for loss of business profits, business interruption, loss of business information and the like) even if ShopBot has been informed of the possibility of such damages.

Use of this product for any period of time constitutes your assumed acceptance of this agreement and subjects you to its contents.

 Copyright 2008 ShopBot Tools, Inc

Getting Started with Your ShopBot Control Software

We know you are anxious to try out your tool. If you just want to learn the minimum and figure it out yourself, read this first section then go give your tool a test drive. You can come back for the juicy details later.

With your ShopBot we provide two kinds of software: The ShopBot Control Software that runs your tool and the ShopBot PartWorks Design (CAD/CAM) Software. This first section of the *Users Guide* describes the ShopBot Control Software. This is the software that you will use for setting up and positioning your tool. Most importantly, you will use it to run the Part Files containing the cutting path instructions for your tool. Think of this software as a "Control Panel" for your ShopBot. This software is also what takes care of all the low level production of movement by the tool. The ShopBot software has many functions and features and we will try and familiarize you with some of these here. You can even do some basic cutting and machining right from this Control Panel. But in making your ShopBot work for you, the ShopBot Control Software is normally just the final step in cutting your Part Files. Most of your planning or design time will be spent with other software such as PartWorks that you will use to create your design, set up your toolpaths (set the order of cut and compensate for cutter waste), and generate your Part Files. You will primarily use the ShopBot Software only to move your tool into location, adjust a few settings, and then start the cutting or machining process of running a Part File.

The need to choose a software approach or method for creating your Part Files is an important issue for CNC or robotics machining. If you are not familiar with designing for CNC or how Part Files are generated, you should read the section of this Users Guide called: "From Your Idea to a ShopBot Part File".

The ShopBot Control Software

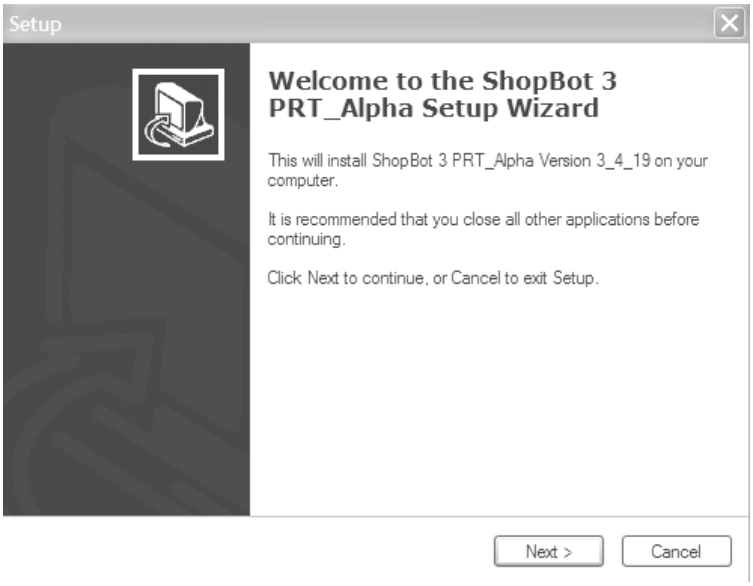
We've tried to keep the basic software interface to ShopBot simple. It will work on a PC running Windows XP Pro or Vista Business or Ultimate (with Vista you will need to turn User Account Control OFF). You will be happiest with at least an 800 MHz processor and 512M of RAM.

You can run the software with or without a ShopBot attached. If you're just trying out the software or testing one of your Part Files, you can use the software in PREVIEW Mode. In PREVIEW Mode you can use almost all features of the software without sending any signals to the tool. You can view the way cuts will be made, verify they will fit on the material, and generally test your Part Files so that you have everything figured out before you get to the shop.

Installing the Software

To install the ShopBot Control Software double-click (run) the program Setup_sb3_x_x.exe from the CD. The most recent version of the install file for the software can downloaded from the ShopBot website (www.ShopBotTools.com) at anytime. After starting the install, the Setup Wizard will launch and you will be greeted with the following Welcome screen. Close any other applications that you have open and click next.

Copyright 2008 ShopBot Tools, Inc


Setup
Welcome to the ShopBot 3 PRT_Alpha Setup Wizard
This will install ShopBot 3 PRT_Alpha Version 3_4_19 on your computer.
It is recommended that you close all other applications before continuing.
Click Next to continue, or Cancel to exit Setup.
Next >
Cancel

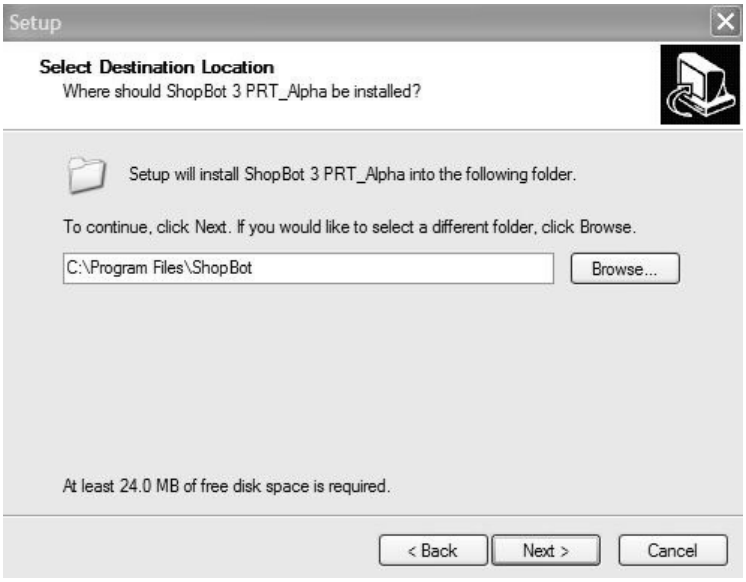
The next screen is the software License Agreement. Use the scroll bar at the right to read the agreement and click on either *I accept the agreement* and Next to continue or click on *I do not accept the agreement* and Next to abort the installation.

Setup
License Agreement
Please read the following important information before continuing.
Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.
SHOPBOT TOOLS, INC.
SOFTWARE LICENSE AGREEMENT
Important - Please Read Carefully Before Accepting
THIS IS A LEGAL AGREEMENT between you, the end user, and ShopBot Tools, Inc. ("ShopBot") which governs your use of the enclosed software program (the "Software") and related written materials (the "Documentation"). By Accepting or by failing to return the Software and Documentation within the
I accept the agreement
I do not accept the agreement
< Back
Next >
Cancel

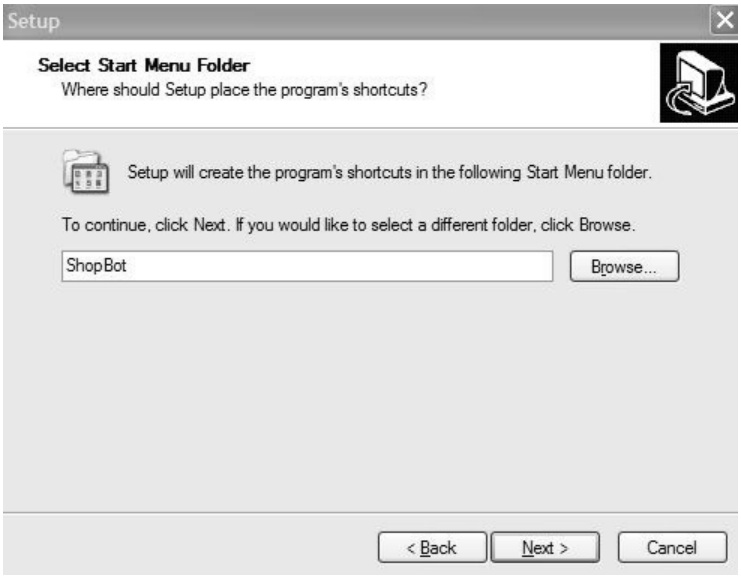
Select the location where the program will be installed. It is recommended that you accept the default location. Click Next to continue

.

Copyright 2008 ShopBot Tools, Inc

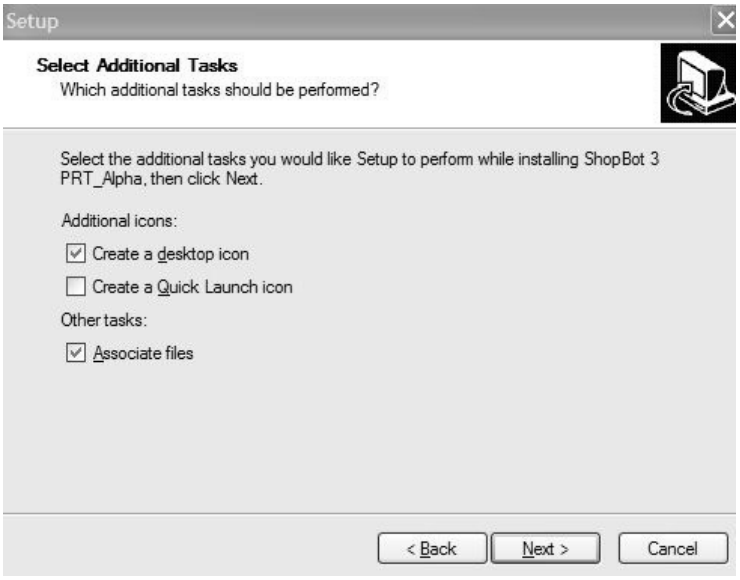

Setup
Select Destination Location
Where should ShopBot 3 PRT_Alpha be installed?
Setup will install ShopBot 3 PRT_Alpha into the following folder.
To continue, click Next. If you would like to select a different folder, click Browse.
C:\Program Files\ShopBot
Browse...
At least 24.0 MB of free disk space is required.
< Back
Next >
Cancel

Select the Start Menu Folder (we recommend you accept the default location) and click Next.

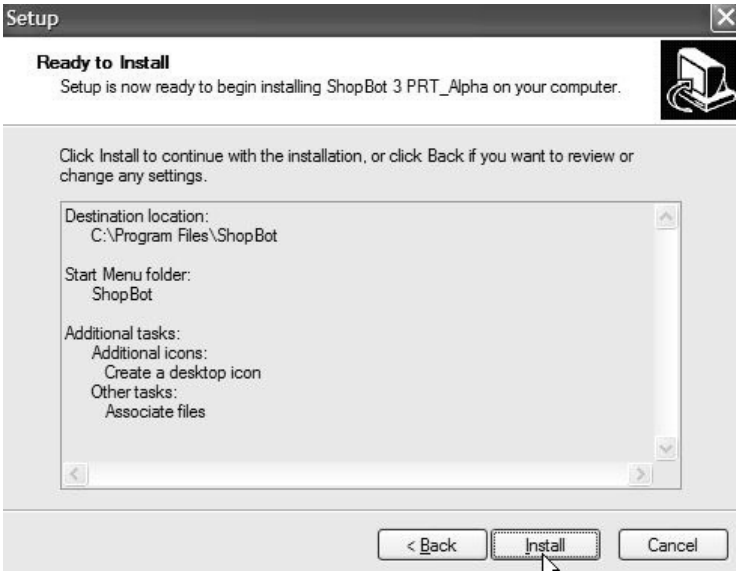

Setup
Select Start Menu Folder
Where should Setup place the program's shortcuts?
Setup will create the program's shortcuts in the following Start Menu folder.
To continue, click Next. If you would like to select a different folder, click Browse.
ShopBot
Browse...
< Back
Next >
Cancel

Select whether you want icons created on the desktop or quick launch toolbar. The *associate files* box is checked by default. Leaving this box checked will automatically open .sbp files in the *ShopBot editor*. Click Next to continue.

Copyright 2008 ShopBot Tools, Inc

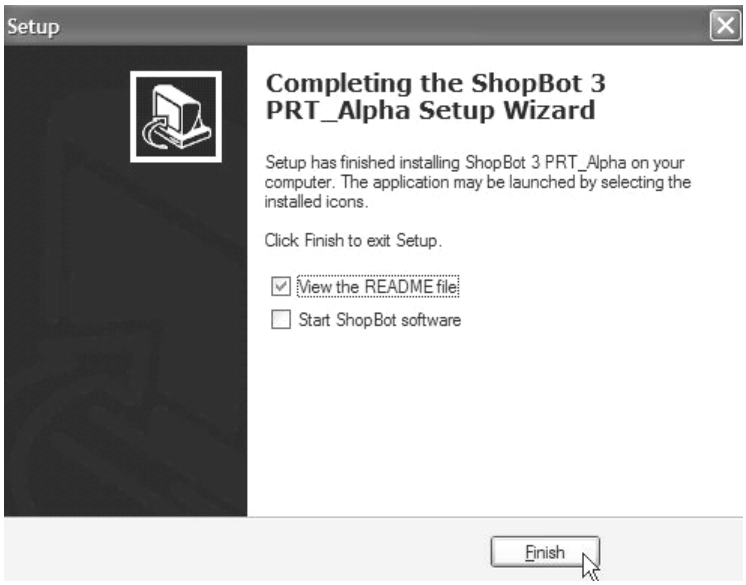

Setup
Select Additional Tasks
Which additional tasks should be performed?
Select the additional tasks you would like Setup to perform while installing ShopBot 3 PRT_Alpha, then click Next.
Additional icons:
Create a desktop icon
Create a Quick Launch icon
Other tasks:
Associate files
< Back
Next >
Cancel

If you are satisfied with all of your settings, click Install to begin the installation.



When the installation is finished you are presented with options for viewing the README file and Starting the ShopBot software. The View README file is checked by default and it is recommended that you read this file before running the software. It contains important information about the latest release.

Copyright 2008 ShopBot Tools, Inc


Setup
Completing the ShopBot 3 PRT_Alpha Setup Wizard
Setup has finished installing ShopBot 3 PRT_Alpha on your computer. The application may be launched by selecting the installed icons.
Click Finish to exit Setup.
View the README file
Start ShopBot software
Finish

ALERT! Make sure that you turn off any sleep mode function or screen savers when you are using ShopBot Software at your tool. Since these functions monitor keyboard action, when you are cutting a large file and no keyboard input occurs, the computer may go to sleep or activate the screen saver and the file will stop executing.

Starting Up

Click on the ShopBot 3 icon to launch the Control Software. The first time the program runs, a warning box pops up that says “Missing SHOPBOT.INI Settings File” Click on “OK” to continue. You will have the option of browsing to the appropriate settings file for your tool.

After selecting the default settings file the software opens and tries to connect to your ShopBot. If your computer is attached to a tool, make sure that the control box is turned on. When the Control Software opens in Move/Cut mode there will be two windows presented on the computer monitor. If you are not connected to a tool, select the box to “Continue in Preview Mode.” A third window is displayed in Preview mode.

Position Window

The red window to the right shows the current position of each axis, the status of the inputs and outputs, current operation mode, and whether movement is in absolute or relative distance. You can run a Part File from this window by clicking the Load Part File button. For safety purposes, when you are in Move/Cut mode this window cannot be repositioned and will remain on top of any other applications.

Control Console

The second window that opens is the Control Console. This window contains all the menu commands for running the tool. You can either click on the item in the menu bar at the top of the window and select the choice in the drop down box or just type the two letter commands that are indicated by underlined letters. Your command selection will appear in the yellow box. An explanation of these commands can be found in the *Command Reference* by clicking on Help. Move and Jog speeds are displayed on the left side of the Control Console. You can change these speeds by typing a new value directly in the box or you can use the menu system to select the Fill-In sheet for speeds, [VS] Command. Several other common settings can be directly entered in the panel.

Preview Window

Copyright 2008 ShopBot Tools, Inc

If you are in Preview Mode a third window appears that depicts your tool's work surface. When a Part File is loaded in this mode the Preview Window will display the toolpath for that file and can be manipulated in several ways for a better view or a simulation of actual cutting of the part. Display options are available in the panels on the sides of the Preview Window. You can also type [VP] to change general settings for the Preview Window. See *Command Reference.*

The Basic ShopBot Command Principle

Whether you are issuing instructions to your ShopBot from the Control Console or from a Part File, there is a standard Command Language for controlling your tool. Each ShopBot "Command" is made up of two letters that are entered sequentially, e.g. [M2] for: "make a 'M'ove in '2'D" (in X and Y axes). The Commands can be selected using the Menu at the top, or you can just type the letters directly, once you've learned them.

In the following discussion, any two letters appearing in brackets are a ShopBot Command, for example [SP], means that you should use the two keystrokes (the letters in the brackets) to activate the function. For example, [SP] is short for "S"et to "P"review Mode -- and you carry out the instruction by pressing "S" and then "P." [SP] sets the software to PREVIEW Mode. Or, you can click the Settings Menu item and then select Preview from the dropdown choices.

Typically, once you've activated the command with the two keystrokes, you'll be prompted for parameters that fully specify the move or action. In the case of [M2], you would be prompted for the new X and Y coordinates for the move. We'll explain the organization of the 2 letters in a moment, but they're designed to be easy to remember and quick to execute, and they make it efficient to operate your tool from the keyboard if you choose. Don't worry; you don't need to remember what each letter means because you'll always have prompts from the screen for whatever letter you need to enter next. In fact, you can just use the arrow keys or your mouse to select Commands if that suits your style better. The two-keystroke system is used so that the moves are executed quickly when you're working at the tool without your having to do a lot of typing or awkward work with a mouse. Moreover, exactly the same commands can be used in Part Files to define complex cuts, patterns, plans, or actions for your ShopBot.

LEAVING WHAT YOU'RE DOING. From any point in your ShopBot activities the Escape key ("Esc") will take you back one step, thereby getting you out of any situation you're in. From the Main Menu, hitting Escape twice will cause you to exit the ShopBot software.

GETTING HELP. The HELP Menu Selection [H] also provides access to an online Command Reference manual with details of the operation of every ShopBot Command -- use the Command [HC] to access this Command Reference Manual. In addition to these HELP functions, the more complicated ShopBot Commands provide "Fill-In Sheets" to help you specify actions you want your ShopBot to take. The use of Fill-In Sheets is described later when we discuss entering parameters for Commands. Fill-In Sheets also show the current value or the default value of required parameters.

Entering Values (Parameters)

Most ShopBot Commands, once activated by entering the two-letter Command or selecting from the menus, require some form of further information that gives additional details for the action (i.e., a value or parameter). For commands like MX or J2 you can enter these values (separated by commas if there is more than one) right on the PARAMETER LINE. For

 Copyright 2008 ShopBot Tools, Inc

many ShopBot Commands, one or more of the indicated values or parameters may be optional. The Command will be executed without your having to provide them. In the case where there is a list of possible parameters, you can leave out any parameter that is not required, but you do need to include commas as place holders up to the last value that you do want to enter.

For example, in Commands involving movements in the both X and Y axes [M2] and [J2]), entering two parameters will produce simultaneous X and Y movement, but you can produce movement in only one axis with a single parameter. “[M2] 3.1, 2” will move the X axis to the location 3.1 and the Y axis to the location 2.0. However, if you just use “[M2] 3.1” you would make an X-axis move to 3.1 inches with the Y axis staying where it is. Moreover, if you first enter a comma, and then a distance, you'll cause movement to the designated location in the Y axis. In this case, "[M2] , 3.1" would make a 3.1 inch move in the Y axis without moving in the X axis. The axis that does not get a parameter stays where it is.

This approach to entering parameters holds for all ShopBot Commands. As long as any required parameter is entered, the Command will execute, with the software using the current settings or appropriate defaults for the other values. To skip over parameters when entering a Command, just enter one or more commas. For example, to set the current Z axis position to a new current value, use the Command [VA; 'V'alues, set 'A'xis location], which accepts four values. To set only the Z axis, use "[VA] , , 1.5" which leaves the X and Y axes and the Accessory axis set at their current values but sets the current Z axis location to 1.5 inches. (NOTE: [VA] defines the current position to the new value; it does not move the tool.) Parameters are entered in the same manner in a ShopBot Part File (with the addition of a separator comma or space after the two letters of the Command itself).

Parameter Line Calculator. You'll notice that the ShopBot PARAMETER line is an automatic calculator. When entering parameters, you may enter simple values like 1.5, or you may enter a complex calculation as a formula, e.g. "1.5 * SQR(4.2)/(23-12.23)." After you've entered the formula, the result will be calculated and automatically substituted in the command being executed. The calculator supports most mathematical and trigonometric functions, see listing at end of this Users Guide (note that functions involving angle are expressed in radians, not degrees). Having a calculated parameter is useful in many situations. It also means that you can easily give your ShopBot fractional locations for moves. For example, to move to 12 and 11/16th inches, just enter the distance as 12+11/16. The 11/16 will automatically be converted to a decimal value before the move is executed. (Note, though, that for negative RELATIVE moves or moves below zero, you would use 2 negatives to accomplish the fractional move; e.g., "-2-7/32").

Drop-Down Calculator. An explicit calculator can also be called up at anytime by entering F10 or [UU] ("U"tilities calc"U"lator). So you can do a quick calculation or make sure that the parameter line calculator supports a function that you wish to use. If you call the calculator with F10 at the point you are filling in the parameter line, you can use the calculator's PASTE function to insert the computed value right into your parameter list. The Calculator functions will make use of any currently stored variables.

Fill-In Sheets. For most ShopBot Commands that have complex parameters, windows that will help you fill in parameters are available. These "Fill-In Sheets" will automatically appear for many Commands. The Fill-In Sheet lays out the various parameters with a more complete explanation of their use than is shown in the parameter line descriptor. You can move between fields in a Fill-In Sheet by tabbing; go backwards with Shift-Tab. When you are finished with a Fill-In Sheet, just hit ENTER and your parameters will be pasted into the Parameter Line and the Command Executed. [NOTE: that you can turn the Fill-In system off if you prefer to enter or edit the parameters manually.]

 Copyright 2008 ShopBot Tools, Inc

As a quick example, enter [CC] to activate the Circle Cutting function. You'll see the Fill-In Sheet listing all the circle cutting parameters. These parameters might also just be entered on the parameter line, but note how the Fill-In Sheet simplifies the process. Also be aware that despite the intimidating number of options for the circle cutting Command, the only required value for [CC] is the circle diameter -- so you can just type in a diameter, hit ENTER to set the Parameters, then ENTER again to cut the circle.
So hit:
CC
6
[ENTER]
... and your ShopBot will make a 6" diameter circle !

So Let's Run a Part File

Now that you have an idea of how to make the software work, let's do the primary thing you will do to be productive and creative with your tool -- run a Part File that carries out a series of your instructions and produces your project. The real potential of CNC and robotics tools is achieved by supplying your tool with a list of instructions to carry out complex cutting and machining movements. That's what's in a Part File and we'll go over it in detail later ... for now, we'll just try it out.

First, let's do it in PREVIEW. Set your software to Preview Mode with the Command [SP]. Then let's run the ShopBot Logo Part File. Start with the Command [FP]. After you enter it, you should get a pop-up window with a list of files in the ShopBot Part Folder (if you are not in the correct folder, you may need to browse to the correct one). Select the logo sample file S_SBLOGO.SBP by clicking on. This is our logo file (you'll note that there are a number of other sample files in the directory, all sample files begin with 'S_...'). After the fill-in sheet comes up, hit ENTER again to accept the default parameters displayed in the Fill-In Sheet and transfer them to the Parameter Line. Then hit ENTER again ... this starts the file running and you will see the file cut in PREVIEW MODE.

In the Preview Window you will see a small ShopBot Logo being drawn down in the bottom left corner. You can zoom in or rotate the view to have a closer look.

Having seen that the Part File looks ok in Preview Mode, switch to move/cut mode [SM] and try an air cutting the same file. During an air cut the router is off and moving above the material so that we can check the tool motion without actually cutting the file. Use the keyboard movement controls ('K' is the shortcut to start them) to move the location of your cutter to somewhere towards the bottom left of the table. The arrow pad display that comes up on the screen should be self-explanatory. The left arrow key, for example, when you push it on the keyboard will move your tool down the X axis towards the 0 location. After you get the tool over to the bottom left, pull the Z axis up several inches above the table. For the moment, let's set this as our home location (0, 0, 0) with the [Z3] Command. Pick the file out again with [FP]. When you are ready to run, hit ENTER twice and your tool will begin executing the logo file. Your ShopBot should move through cutting the logo. This file uses a variety of moves so it is a good initial test of your setup … the file is also an instructive one to study as a Part File example.

If you have a bit in your router, you can also go ahead and cut this file. It was designed to be cut with a 90 degree V-bit as indicated by the instructions in the file (look inside the file with the ShopBot Editor, bring it up with the [FE] Command). Position a piece of material on your table and make sure it is securely held in place with either clamps or vacuum. If using clamps make sure that clamps are out of the path of movement of the tool. Move the cutter to the position on your material where you want the cut to start. Lower the Z axis until the

 Copyright 2008 ShopBot Tools, Inc

bit just touches the work surface. Let's set this as our new home with the [Z3] Command. Alternatively you can type [C2] and [C3] to home the tool if you have a zero plate and proximity switches.

Ready to Cut? Make sure everything is clear. Start the router. Now run the Part File using [FP]. You should see a neat little ShopBot logo carved into your material. The size of the logo can be scaled up or down with the 'Proportion' factor in the [FP] parameters (an X proportion of 2 and a Y proportion of 2 will double the size of the logo). Congratulations on cutting your first ShopBot Part!

... and an additional reminder ....

=== PANIC ! PANIC ! PANIC ! ===

When the tool is executing any movement, the SPACEBAR becomes a STOP BUTTON. >>>> HIT the SPACEBAR and ALL MOTION OF THE TOOL WILL STOP. You will be prompted for what to do next. At very slow speeds, the tool will stop immediately. At higher speeds, it will make a smooth stop where the movement speed is ramped down and the cutter pulls up. Even at slow speeds a smooth stop can be forced using the 'S' key for STOP. The threshold for fast or slow stops is set in [VR].

Your tool is also equipped with an Emergency Stop button. This button will cut power to the motors and router on PRSalphas, but will only stop movement of the carriage on PRSs.

Copyright 2008 ShopBot Tools, Inc

Getting Help

With every manual, there is a laminated Quick Reference Guide that has a summary of the ShopBot Commands, and the reverse has a list of the parameters for each command. It is a very useful document for simple questions.

If you are having mechanical or electronic problems with your ShopBot, and have exhausted the suggestions here in the troubleshooting section of this User Guide. We suggest you send Tech Support an email or call (we are always available during business hours. After hours and on holidays we check for messages every few hours and will do our best to get back to you). Our diagnostic website, http://drshopbot.com, may also help you get your problem solved.

If your questions are related to making things with your ShopBot, you will want to visit the ShopBot Forum on our web site for ideas:
http://www.shopbottools.com

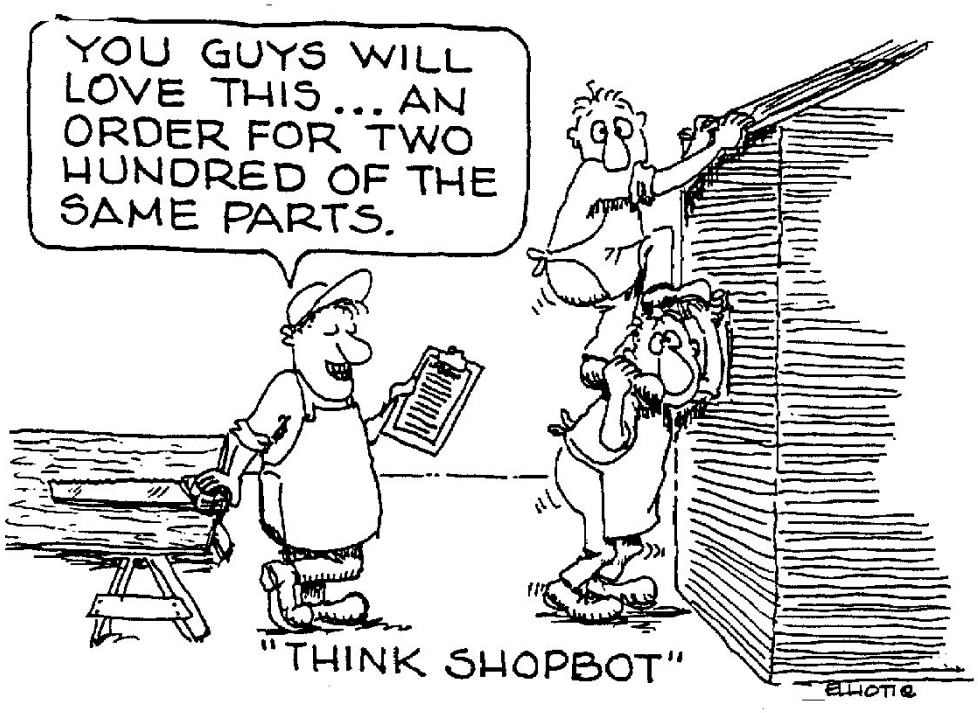
Whenever you communicate with us about a problem, it is useful to us if you (under Help in the ShopBot software is a function to Send a Tech Support Request that will do this automatically if you have an internet connection on the computer):

- Provide all the detail that you can about the exact steps that led up to a particular problem
- E-mail us a sample of any file that may be related to the problem or illustrate the problem
- Send along the file named PROBLEM.LOG that you will find in your ShopBot Folder/Directory (this file allows us to re-create your exact set-up).

ShopBot Support is available:

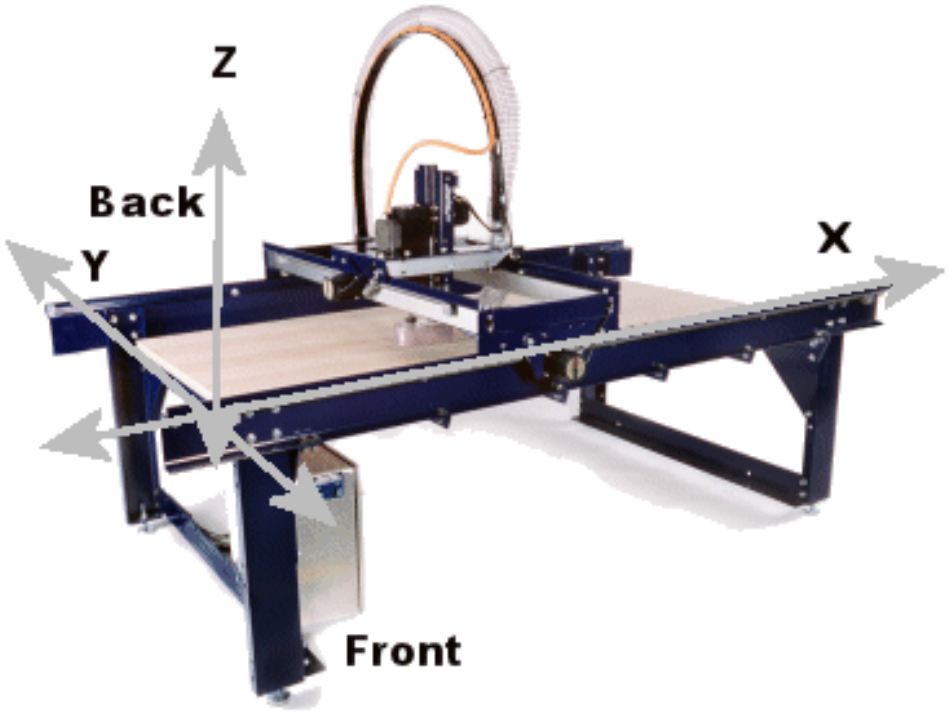
BY email: support@shopbottools.com
BY fax: 919 680-4900
BY phone: 919 680-4800

Copyright 2008 ShopBot Tools, Inc



The Layout of Your ShopBot Table: Directions, Distances, and Base Coordinates

Here we take up how you should think about the layout of your work area and the different ways of moving around in it.



Copyright 2008 ShopBot Tools, Inc

The Logic of XYZ

It sometimes takes a little while to get comfortable with the direction system that we use to move your ShopBot around. One of the easiest ways to think about it is to just imagine that the table is a big piece of graph paper. Have a look at the orientation drawing above (this is similar to the one in the Assembly Manual. If you have a Buddy tool the same direction scheme applies when facing the tool from the right side of the gantry. From your position at the front of the tool the graph paper is spread out in front of you. X and Y are on the flat table. X is the long axis from the left to right of the table (the short, table-axis on a Buddy). Y is the short axis that goes across the gantry. Low values are at the bottom left. Values increase in the X axis as you move to the right. Values increase in the Y axis as you move to the far side of the table. Any location on the table is specified with an X and Y location value, and these are the values we use as instructions to move the tool around. Whether you are making a move at cutting speed, for example an [M2] move, or a move at jog (rapid positioning) speed, for example [J2], the two parameters that follow the Command are the X and Y locations that the tool will move to.

The Z dimension is the third axis, the perpendicular or up-down movement of your tool. Increasing values with an [MZ] Command will move the Z axis up; decreasing values will move the Z axis down. For convenience the top surface of your material is often set to a Z value of 0 (that is, the tip of the cutter would be just touching the top of the material). Negative values then take you into the material; positive values pull the cutter above it.

Giving Directions; ABSOLUTE vs RELATIVE distance

We just described the location system in terms of "absolute" locations. Your tool is always considered to be at some specific location in X, Y, and Z and this is the location that is displayed on the screen. Most frequently, you or a Part File will instruct your tool to move by just giving it an instruction to go to a specific X, Y, Z location. For example the instruction [M2] 3, 4.5 tells your tool to go to the location 3 in the X axis, 4.5 in the Y axis; [M3] 5, 6, -.5 would send it to an X value of 5, a Y value of 6, and a Z depth of -.5. We call this an ABSOLUTE Distance instruction since you specify the exact location to which the tool moves. Such a move might be a cutting move with the router running through the material, or it could be a positioning move with the cutter above the material during the motion. Your ShopBot can also make RELATIVE Distance moves. To move Relative, you place the tool in RELATIVE Distance Mode with the [SR] Command. In Relative Mode all moves are incremental moves, for example the instruction [M2] 3,-1 would tell your tool to move an additional 3 inches in the X axis from its current location, and back or negative 1 inch from its current location in the Y axis. The way the software interprets move instructions can be changed at any time with the instruction [SA] to set up ABSOLUTE Distance and [SR] to switch to RELATIVE Distance. The distance system can be switched back and forth inside a file, or when you are working your ShopBot from the keyboard.

Generally speaking, we find it best to use ABSOLUTE Distance in Part Files. This avoids any accumulated errors that might arise from mistakes or rounding. It also makes it possible to look at a line in a Part File and know where the tool should physically be at that point in the file ... note, that if you have a File with a list of RELATIVE Distance Commands the current location of the tool is not obvious from reading a single line because the move series would have to be added up to know where you are or should be. For working at the keyboard, either type of Distance Command works fine, but for many users always working in ABSOLUTE MODE keeps life simpler.

Copyright 2008 ShopBot Tools, Inc

You Can Reorganize Your "Graph Paper"

In the next few sections we're going to add a bit more power and flexibility to this system of locations and distances. This is going to seem a bit confusing at first, but it's going to make things a lot easier in the long run.

At anytime it is easy to reorganize where the center point (0,0,0 or "home") of the conceptual graph of your tool's work area is located. Most of the time, you will probably want to keep the 0,0,0 location at the bottom left of the table to avoid using negative numbers. But if you happen to be using a smaller work area than the full table, you may want to move your 0,0,0 to some other position in space. For example, you might want to work with 3' x 3' sheets on the right side of the tool. Moving your working coordinates is easy. Just move the tool to the place you want the new zero location (e.g., drive it over with the arrow key control activated by [SK] or the shortcut 'K'), then use the Zeroing Command for the axes you want to zero. For example [Z2] would define the current location as the new working 0,0 for the X and Y axes; [Z3] would zero the X,Y, and Z axes. You can move the zero locations around as frequently as you would like in order to make your work on the tool more convenient.

There is an additional method to redefine the location of the axes, the [VA] Command. This Command is used to set any or all current axes locations to a specific value. For example, let's say the tool is currently at the XY location of 20, 10, and that I want to change the location of the X and Y axes so that this current location becomes 10, 15 (i.e., shifting of the X axis zero location 10 inches to the right and the Y axis zero down by 5). I would use [VA] 10, 15 to redefine the current location value to these new values ... which at the same time shifts the X and Y axes and their zero points to reflect this new current location. Using [VA] 0, 0 has the same effect as issuing the [Z2] Command.

Temporary Zero Points When Cutting Files

ShopBot provides an additional method of manipulating the home or zero location when you are running a Part File. When cutting a Part File you can establish a temporary zero point just for cutting the file. You will probably use this method frequently when working with Part Files. It is called the File "Offset" function and is one of the parameters available when running a file. If a Part File is run in "Offset" mode, the current location of the tool at the point where the file is started will be taken as a temporary zero point for executing the file -- the file will thus be "offset" from this starting location. This means that even though you created the Part File using absolute coordinates, you can position it anywhere in your table coordinate system to cut it. When you run the Part File and set up the offset, you can specify that the offset will occur in 2-D or 3-D. 2-D means the X and Y will be offset from the current location but the absolute values in the file will be used for the Z distances; 3-D offset means movements in all the axes will be offset from the starting point. You will typically use 2-D offset because you will want to position a part in X and Y, but keep the Z relationship constant with respect to the table.

An example of cutting a file using the offset instruction would be as follows (note that you've probably moved the tool out to some appropriate starting point on the table). Then to cut the file, you create the Command: [FP] MYPART.SBP, 1,1,1,1,2 Here the last 2 is the offset in 2D instruction. The 4 1's are for X,Y, and Z proportion and number of repetitions -- they could have been left out in the Command as long as we included the commas. The Command can be set up like this by just selecting the 2-D Offset in the fill-in sheet and hitting <ENTER> twice.

One of the most useful features of the ability to 'offset' parts is for placing multiple parts on a single sheet. The offset feature makes nesting a number of copies of the same part on a sheet convenient. The simplest way to do this is to first create a Part File for the part itself.

 Copyright 2008 ShopBot Tools, Inc

Then create a second "master" Part File that moves the tool to each starting location and then executes the [FP] instruction to cut the part in offset mode. An example of nesting in a Part File follows (if you aren't yet familiar or comfortable with Part Files, look at this again after you've read the Part File section further on):

```
'Start of master Part File ALLPARTS.SBP
M2, 10, 4                        'move to location to start first part
FP, MYPART.SBP , , , , , 2       'execute with 2D offset (used commas here)
M2, 20, 4                        'move to location for start of second part
FP, MYPART.SBP , , , , , 2       'cut 2nd with 2D offset
M2, 30, 4                        '... and so on
FP, MYPART.SBP , , , , , 2
M2, 40, 4
FP, MYPART.SBP , , , , , 2
M2, 50, 4
FP, MYPART.SBP , , , , , 2
M2, 10, 8
FP, MYPART.SBP , , , , , 2
```

Now that you have mastered the art of using the 2-D offset to nest files (or especially if you cannot seem to get the hang of it) the Control Software has a file that does all the programming for you. The file is S_Nest.sbp and is located in the SBParts folder with the other sample files. Call up this file with the [FP] command and follow the prompts to create your nesting file.

Table Base Coordinates vs Current Working Coordinates

Sometimes it is useful to be able to temporarily redefine a location on the table, calling it zero or giving it a new value -- and yet, after some task has been carried out, to revert to the original coordinates. In operations where you do not want to lose the exact home (0,0,0,0 in X, Y, Z and Acc) position of your tool because of operations like part registration or other jigging procedures, it is particularly important to be able maintain a set of Base Table Coordinates that represent real physical locations on your table.

TIP: The following section on the Base Coordinate System may sound a bit esoteric and/or needlessly complex. You can run your tool just fine without ever worrying about the Base Table Coordinate system. Feel free to ignore this system until you have you have a real need for it. However, if you want to do Software Limit Checking (next section) you will need to set up your Base Coordinate System.

ShopBot provides two Commands to set up and utilize a Base Coordinates system. The Command [ZT; 'Z'ero 'T'able base coordinates] is used to set the Table Base coordinate system to 0,0,0,0. Typically you would move your tool to the physical 0,0,0,0 location that you want to establish (you would only worry about the 4th axis if you use an accessory axis). You would probably want to initially Zero this location in terms of your working coordinates with the command [Z3] or [Z4]. Then you would use [ZT] to set the Table Base Coordinate system to 0,0,0,0. The point where your tool is physically sitting when you type [ZT] will now be considered its Base Home or Base Zero location. From this point, you can change the working location of your tool with the Axis Zeroing Commands such as [Z2] or with the [VA] Command and the Base Zero will not be lost. Even though the display changes to reflect the new working locations that you have entered, the Table Base Coordinate system location will be maintained. At any point, you can use the Command [ST] to leave the working coordinates and reset the location of your tool to its location in terms of Base Coordinates. {Viewing the current Values using [UV] will also tell you, at anytime, where your tool is in relation to the Base Coordinate system (i.e., in the Base Coordinate Location line part way down the page). For those into very technical manipulations, you can

 Copyright 2008 ShopBot Tools, Inc

change the Base Coordinate locations by using the 5th - 8th parameters of [VA], see the Command Reference manual. }

Software Limit Checking

Your software is set by default to check each location you send it to in order to determine if that location is within the limits of your table. The Command to turn on and off this verification testing is [SF;'S'et 'F'ile and Move Limit Checking]. Limit checking is done in terms of the Base Coordinate system. Thus, if you want to use the limit checking, you will need to make sure that you keep your Base Coordinate system current and oriented to the true layout of your tool. If you have not updated the Base Coordinate system, you may get an incorrect Limit Check report. In this case, you will have the option of disabling the limit checking and continuing. Note that using the C2 and C3 Zeroing Commands will automatically set your Table Base Coordinates correctly.

You can use software limit checking in conjunction with PREVIEW MODE viewing of a Part File as a verification test that all the moves of a particular cutting file will be within a specified region. You can use [VL] to set up the values for limits in each axis that will be tested. Note that the default settings are for 96” X 48” ShopBots. For BenchTop Tools or custom sized tools, you will need to set your table limits with the [VL] Command.

Maintaining Accurate XYZ Locations

You should be aware that both the Table Base Coordinates and Working Coordinates remain accurate only as long as your tool is not manually moved. Your tool can be manually moved either by pushing it with the motors off, or by having it accidentally pushed when the computer and software are not operational. When the software is running normally, the motors are on and the tool will be securely locked in position. However, when the motors are off, it is vulnerable to being inadvertently moved and there is no way for the software to know that this has happened. Thus, you should be careful that your tool is not pushed or jarred when it is out of use, if you are trying to maintain your location from session to session.

Your can use the [C3] Command to Zero the X and Y axes anytime the power has been off, or after any operation where you have moved your Zero point. You can use the [C2] Command and the Z-Zero plate supplied with your tool to Zero the Z axis to either the top of the table or the top of your material.

Changing the Direction of Movement of an Axis

It is possible to change the direction that an axis moves. This change can be made in two ways. If the direction that an axis is going is not what you want, and you would like to permanently change it, then the best thing to do is to follow the directions in the Troubleshooting section at the end of this Guide or at the end of the Assembly Manual on changing motor wires to change the direction of an axis. **WARNING: This only applies to PRS <u>standard</u> tools. Do not attempt to change the wires on PRSalpha ShopBots or you will void your warranty.** Only two wires need to be switched, and it's an easy change to make. Alternatively, for a short-term change in direction (for example, to change the direction of plunging or to create a mirror image of a part) direction can be reversed with the software by changing the sign of the Unit Value for the axis using the [VU] Command. So, for example, to make your Z axis go down with increasing positive values instead of the default negative plunge, you would change the unit setting from 2000 to -2000 ([VU] , , -2000). With some ShopBot functions, using a negative proportion in the parameter settings will also produce movement in the reverse direction for an axis.

Copyright 2008 ShopBot Tools, Inc

Software Axis Designation

You have the ability to assign any axis to any channel in the Control Software. The default configuration assigns channel 1 as X, channel 2 as Y, channel 3 as Z, channel 4 as A and the fifth channel as B. There are a number of reasons that you may wish to change these default settings. For example, you may have set your table up in a way that requires you to switch the X and Y axis. This change can be accomplished in the Control Software. Type [VI] and in the fill-in sheet, switch the channel 1 and channel 2 assignments and hit Enter twice to save your changes.

Optional 4th or 5th Axis or Channel

The ShopBot Software supports an optional 4th or 5th axis or channel, which can be configured as a second Z axis or as some other type of movement system (e.g., an indexer). The 4th axis is called the Accessory axis in the software and its default settings are the same as those for Z axis. If this axis is used as a second Z axis, then it moves in the vertical axis just like the Z axis. Typically, an optional Z axis will be aligned in the Y axis with the first Z axis, but at a fixed distance from it in the X axis. The offset can be measured and used to compensate in Part Files so that both axes location can be known and controlled. The 5th axis is designated as the B axis. In order for these additional axes to be available to you in the software type [VD] and change the Number of Axes to 4 or 5.

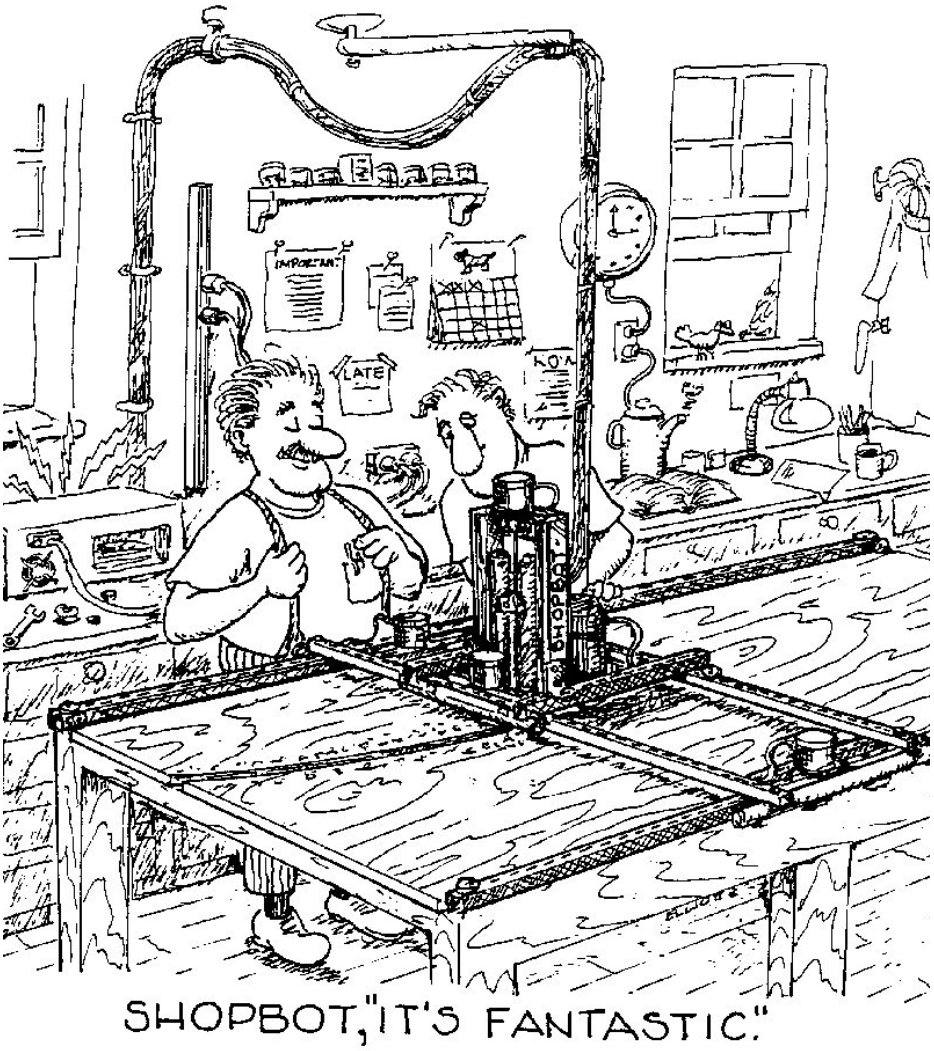
Inches or Millimeters

You can use your ShopBot in inch units or in millimeter units, and you can switch back and forth between the two. The unit system in use is set with the third parameter of the [VD; 'V'alues 'D'isplay] Command. Entering the Command [VD] , , 0 sets your ShopBot to run with a display of 'inches' (the default setting). Entering the Command [VD] , , 1 sets your ShopBot to run with a display of 'mm'. The type of measurement unit that is in use can be found by clicking on the Other Common tab on the Control Console. You can also easily switch between units on this tab.

When you shift between the two types of display, say inches to mm, you will change the way that all units are displayed on the screen and how the numbers entered are interpreted in movement commands. All current settings and values that are stored as distance measurement units are also converted. All Move, Jog, and Ramp Speeds are changed to their metric equivalent as well as other Values. For example, the Cutter Diameter value (normally set with [VC]) would be converted from its inch value to its metric equivalent. Because all stored values which are based on distance are converted when you switch from inches to mm or vice versa, you need to make certain that whenever you enter a value that you enter it in the correct measurement system. This means that if you are working in metric units, you must enter values such as Move Speed and Cutter Diameter in appropriate metric units.

ALERT! ShopBot's Unit Values, which are set with [VU], also depend on the measurement system in use. The table provided at the end of this User Guide giving the Unit Values for different gear ratios, is correct for working in inches. If you switch your tool from inches to mm your current values will automatically converted. However, if you change these values when in the mm display system, you need to enter the values in their metric form. It's probably best to always enter the correct Unit Values for your tool while in the 'inch' system, and then switch to metric display.

Copyright 2008 ShopBot Tools, Inc



What Your Tool Can Do

In this section of the User's Guide we describe the general types of ShopBot Commands. The Quick Reference sheet included with your documentation provides short descriptions of each of the ShopBot Commands. The full details of all Commands and their parameters are provided in the Command Reference manual. The Command Reference is provided here in hard copy, and is instantly available in the ShopBot Software by hitting the Command letters [HC; 'H'elp 'C'ommand Reference]. In all our documentation, Commands are organized according to the format established earlier. The Command is indicated by the two bold capital letters, with the command name following the dash. Parameters are shown in italics on the following line. If there are no italicized terms after a Command, then it takes no parameters and the Command will be executed immediately after the second Command key is entered. Note that in using ShopBot Commands, distance parameters can be either Absolute or Relative depending on the current Distance setting of the software.

Some Commands have parameters that are required for the Command to function; in our documentation these parameters are shown outside the braces ({ }). As few or as many optional parameter as desired may be used; optional parameters are inside the braces. In most cases optional parameters have default values which will be applied when the Command is executed. When entering parameters on the command line or in a file, they must be separated by commas. If a parameter is skipped, a comma must still be entered as

 Copyright 2008 ShopBot Tools, Inc

a place-holder if a subsequent parameter is to be provided. For example, to set the location of the Z axis to 1.5, the [VA; 'V'alues 'A'xis location] Command would be followed by: , , 1.5 because we are just providing place holders for the X and Y locations:
[VA] , , 1.5

Note that in Part Files, the two-letter ShopBot Commands which take parameters must always be followed by a separator, either a comma or at least one space (e.g., 'M2, 3.1,2.43' or 'M2 3.1, 2.43').

Giving Action Commands

You've probably heard the old adage that a robot can't do anything that it is not instructed to do -- it's true. But instructions can be very powerful tools, and instructions are what making the most productive use of your ShopBot is all about. So let's get started working on how to tell your ShopBot to do things...

By now, you have probably realized that Commands entered in the ShopBot Software let you do at least two basic kinds of things: First, you can instruct your tool to carry out some action, for example make a move or cut a file. And second, you can change something about the way the tool works, for example change the value of its current location, the speed the tool moves at, and so forth. Let's consider these Command types in a bit more detail.

In terms of actions, you can tell your tool to do something like make a basic Move or Jog speed move. Or you can tell your tool to carry out one of the built in Cutting routines for circles, arcs, and rectangles. Or you can tell your tool to cut a File. These "Action" Commands typically have required parameters and the parameters for the File and Cut Commands have much in common. Many include the option to alter the proportion or scaling of each axis as well as specify the number of times a Command is repeated. You always provide parameters after you enter the two-letter Commands.

Changing Settings and Values (Commands)

A second type of Command sets up the way your tool works and allows you to configure it best for the task at hand. We've organized the things you do to define tool operation in a logical way. One of the things you will do frequently is to Zero an axis. Thus there is a Menu category just for zeroing that lets you choose exactly what you want to zero. Note that there are no parameters with the Zero Commands. They execute instantly as soon as you hit the two letters (so you should be careful using them).

There are a number of ShopBot settings that are either ON or OFF, or are toggled between two states or conditions. RELATIVE vs ABSOLUTE Distances is a good example. We call features of this type "Settings". Settings are activated or switched with just a single Command. So for example, the Distance setting is done with either the Command [SA] or [SR]. Most other Setting Commands are similar and require no parameters because a simple change in operational mode or condition is being carried out.

On the other hand, many ShopBot functions require that a number of details of operation be specified. We call these various details of function "Values". Commands in the Values menu allow you to provide parameters to specify how your ShopBot will behave. Here, for example, you define speeds or feed rates [VS; 'V'alues 'S'peeds]; acceleration profiles for how quickly your tool ramps speeds up and slows down [VR; 'V'alues 'R'amps]; the size of the cutter you are using, which will be used to compute cutter compensation in built-in circle and rectangle routines [VC; 'V'alues 'C'utter]; distances that will be used with the

 Copyright 2008 ShopBot Tools, Inc

"tabbing" function [VB; ‘V’alues ta’B’bing]; how the TableTop Preview screen is configured [VP; ‘V’alues ‘P’review]; and many other aspects of how your ShopBot functions.

All of the current Settings and Values of your tool are continuously saved to the hard disk on your computer. This means that whenever you make changes to these settings, the changes will be preserved when you restart your ShopBot the next time. Settings are saved in a file called SHOPBOT.INI. This file is a "text" file that you can read (or modify) with a text editor (such as the text editor called up by ShopBot). This file is read whenever your software is started. It is automatically saved after every Command you enter.

TIP: If your ShopBot is behaving erratically and you feel that the data in the SHOPBOT.INI file may have become corrupted, simply type the [UR] command and follow the prompts to select your default file. In some cases it may be necessary to manually delete the ShopBot.ini file from your ShopBot 3 folder. The next time the ShopBot software is started a new SHOPBOT.INI file will be created with default settings after you select the appropriate .sbd file for your tool.

TIP: If you would like to save current Settings and Values for re-use in a particular situation or with a particular Part File, you can use the [US] Utility to save the current Settings and Values to a File that you specify. Then run this file with [FP] to re-establish the saved Settings and Values any time that you need them. This is easiest if you name the file with an .SBP extension.

ALERT! When the SHOPBOT.INI file is first created its parameters are derived from the default Settings & Values file for your tool. These files have a .sbd extension and there are separate files for different sized tools. They are located in the: C:\Program Files\ShopBot\ShopBot 3\Settings folder.

Utility Commands

The ShopBot Control Software provides a number of additional functions and Utilities. These functions are also accessible from the category menu and provide the capability of Recording Commands that are entered from the keyboard as well as functions for viewing the current settings of the tool and software [UV; ‘U’tilities ‘V’iew values] and providing diagnostic information [UD; ‘U’tilities ‘D’iagnostics]. The calculator can be accessed through the Utilities Menu category [UU]. Remember that there are also Help Files available from the Help Menu [H].

ShopBot Part Files (.sbp)

You could control your ShopBot tool entirely through keyboard input using the Commands like those we have been discussing. For some small projects, this may be all that you need to do. However, the real potential of CNC and robotics tools is achieved by supplying your tool with a list of instructions to carry out complex cutting and machining movements. Such intricate cutting capability is why a robotics tool can do things you can't do on your own -- and it can do them hundreds or thousands of times without a glitch.

What's in a Part File

The list of instructions is provided through Part Files that you write or generate in other software and bring to ShopBot. The list of instructions looks just like the list of Commands you would enter from the keyboard. On each line of the list, there is a two-letter Command followed in most cases by parameters. The lists are saved in a file and we call these Part Files. These files are what are referred to as text files. They can be created, viewed, and edited with a text editor (simple word processor).

 Copyright 2008 ShopBot Tools, Inc

Much of the time, your Part File instructions will just be a list of the moves needed to cut your part. A Part File can be a short and simple list that looks like the following:

```
JZ,  .5
J2,  10, 10
MZ, -.25
M2,  20, 10
M2,  20, 5
M2,  10, 5
M2,  10, 10
JZ,  .5
```

Here we have instructed your tool to cut a rectangle. First (JZ, .5), we pull the bit up to make sure it is not in the material when we make our move to the starting point (typically, you set up your Z axis with the convention that the top of your material is the Z location 0, so we're pulling it half an inch above the material). Next, the tool will Jog at positioning speed to the XY location 10, 10 (we will assume that the ShopBot is configured for Absolute Distance moves). This is our starting point. Then the bit is moved down to -.25 (MZ, -.25) at Move speed now because we are cutting into material); it will drill into the material and be ready for our cut. Then, in this file, there are 4 cutting speed moves (M2's) that will define a 10 x 5 rectangle, with the starting point being in the upper left corner of the rectangle, and going around the rectangle in a clockwise direction (if you were using a 1/4in bit, this would create a cut-out rectangle that is actually 9.75 x 4.75 because the center of the bit would be following the specified dimensions, and thus the part is smaller). Once the part is cut, we pull up out of the material with a JZ, .5 jog up. This Part File illustrates the cutting of a simple part.

You could also have cut this same rectangle using the built-in 'Cut Rectangle' feature in the software [CR]. You could do it from the keyboard, or create a Part File that uses the [CR] feature. The file for doing it this way would look like:

```
JZ,  .5
J2,  10, 10
CR,  10, 5,  ,  ,  , -.25,  ,  , 1
```

Here, after moving to the starting point, the [CR] function makes the rectangle and takes care of plunging (note the 'plunge from Z location 0' option is set with the last '1' in the instruction line, which means that the Z axis moves to 0 before executing the -.25 plunge) and pulling back up at the end of the cut. Additionally, if we had wanted a rectangle cut out that was the actual specified size, we could have set the third parameter to "O" and we would have gotten a cut-out with the actual outside dimensions of 10 x 5 based on the known cutter size, which is set with the [VC] Command.

Using the Editor

The files that ShopBot reads are in what is called 'text' format. That means you can look at them and edit them with any text editor or word-processor, however we recommend using the text editor inside the ShopBot software which is activated with [FE]. If you do use your own word-processor or an editor like the Windows NotePad, make certain that after you have created or modified a Part File that it is re-saved in generic 'text' format and not a special format of your word-processor. The file name extension for a ShopBot Part File is '.sbp' . (Other ShopBot related files have file extensions that start with '.sb_' and end with another letter for descriptive convenience.

Copyright 2008 ShopBot Tools, Inc

The [FE] Editor has all the functions you will ever need and it will allow you to conveniently access your Part Files in a context in which you can use the various ShopBot Help functions. Start it up by hitting 'FE' and selecting a file name of the file that you want to edit type [FN] to open a blank editor page for creating a new file. You can type in a series of Commands like those in the paragraphs above, then save them with 'Alt-F; S' to a file name that you designate. You close the Editor with 'Alt-X'. Now you're ready to execute the file, or cut the part, using [FP].

TIP: So that you will have easy access to the Editor and to ShopBot preview functions, it may be convenient to install the ShopBot software not only on the computer that runs your tool, but also on the design computer at your work desk in the office. Then you will be able to write, edit, and preview your work from the comfort of your office chair.

Automatic Recording of Things You Do at the Keyboard

Now you have the basic idea of a Part File and how to create one with the Editor. The list that the file provides can be short and sweet, as in the above examples. In this case, you could just 'write' the instruction list yourself. For some simple projects, this will be the way to go. You also have the option of recording a series of Commands as you enter them right from the keyboard and then further editing your list of instructions using the Record Category functions.

Here's how 'Record' works: As you are entering Commands at the keyboard your ShopBot software is usually recording them. This allows you to go back and replay a series of things that you did, or allows you to specifically create a cutting routine by entering commands at the tool -- either in Move/Cut or Preview Mode. The recording system is quite flexible. You can play back a part of what you've done ([RR]), or you can save it to a file and edit it ([RS]). You can erase what you've done and start over ([RZ]). And, you can temporarily suspend recording while you have the tool do something unrelated ([RI]). By default, and as indicated by the little squiggles in front of the 'R' on the Menu, Recording is normally On.

To create a file using the Record functions, you would typically start by erasing everything in the 'record memory' with the [RZ] Command. Then you would enter the Commands you wanted, watching as the tool (or Preview Display) went through the actions you were developing. You could temporarily suspend recording and restart it, using [RI] to inactivate and then [RA] to activate it. Then when you were finished, you would use the [RS] Command to save/edit a file. Your series of Commands will be displayed to you in the Editor as part of a new Part File created by the Record function and you can do any additional editing or fine tuning that is needed before leaving the Editor.

In addition to making a record of commands, you can also record positions that you move to in the keyboard Arrow-Key mode and then save these positions as a move from the previous location. You can choose what type of move to use (2D or 3D) and whether to save it as a Move Command or a Jog Command.

Don't Let this Scare You...

Part Files offer virtually unlimited control of your tool and the execution of cutting instructions. There are some additional instructions that also can be included in a Part File to help control or program how the Part File runs. We'll mostly save discussion of these for the Part File Programming Handbook because you don't need to know anything about them to understand the main principles here. These additional instructions make our Part File system into what computer types call a 'programming language'. We don't want to worry you about this, you may never want to think about 'programming' and you probably won't need to unless you are interested in getting into it. But there are just a few things that you may want to know that will help you out with day-to-day tasks in Part File work without

 Copyright 2008 ShopBot Tools, Inc

getting you deep into programming. Of course, realize that if you've written a single line of a Part File ... you are already a programmer!

Remark Lines and Documenting Your File. It is very helpful when creating Part Files to make notes for yourself in the file so that when you look at it at some later time, what you've done will make sense to you. This is a process that programmers call 'documenting'. [There is at least one school of programming which insists that you can not over-document a file ... so at least consider making enough notes to yourself that you'll understand what you've done at a later time.] To make a note to yourself in a file, all you need to do is type in an apostrophe (') and everything that follows it on that line will be ignored by the ShopBot software when the file is executed. These non-executing lines or phrases are usually referred to as 'remarks' or 'comments'. You can put in the apostrophe and make a note to yourself about what a particular line, or portion of the file, is supposed to be doing. You can put the apostrophe at the beginning of the line and use the whole line for comments, or you can put it after a Command and make a short note to yourself. If you want to make several lines of comments, put an apostrophe at the beginning of each line. The three letters 'REM' (for Remark) can be used instead of an apostrophe to begin a full-line comment. A full-line comment is displayed at the bottom of the screen when a file is executing, so this is an easy way to remind yourself of things while the file is being processed (see the 'PAUSE' instruction below to learn how to make this display remain visible for as long as you would like).

```
'This is an example of documenting a file
'You can use an apostrophe (or a REM at start of a line)
REM You can use a REM (or an apostrophe at start of line)
'This file creates a rectangle (written by me, 1/1/2000)
‘ ... None of the above effects the cutting of your file
‘ ... The line below is left blank for legibility

JZ, .5                    'this line pulls the bit up
J2, 10, 10                'you can add a comment to line
CR, 10, 5,  ,  ,  , -.25,  ,  , 1
      'You can indent lines or comments and add blank lines
'The End
```

The 'PAUSE' Instruction. The 'PAUSE' Programming Instruction is one function that you will find useful even if you don't get deeply involved with the programming features of ShopBot. Using this Programming Instruction -- that is, putting it in a line of your Part File -- allows you to put a brief pause or a stop in the action of executing the Part File. If you put a PAUSE on a line, it will cause execution of the file to stop, until you hit a key on the keyboard. If you put a PAUSE followed by a number, say 'PAUSE 3', there will be a 3 second pause in the action and then the file execution will continue. If you put a comment line with an apostrophe just before the pause, that line will be displayed during the time the PAUSE is in effect. For example, you might use the PAUSE as follows:

```
'Time to Change Bits  .... (Hit Any Key when Ready to Continue)
 PAUSE
```

When your program gets to this place in the file, the message to Change Bits will be displayed at the bottom of the screen and your ShopBot would stop until a key is hit.

Adding Setup Lines to the Start of a File. After you have created a Part File that does some type of cutting or machining for you, you will want to consider putting a few setup lines at the beginning of the file, particularly if this is a file you will use regularly. Because all ShopBot Commands can be used in a Part File, you can put all the instructions that you

 Copyright 2008 ShopBot Tools, Inc

might want to give to configure your tool at the start of the file. For example, to make sure that your tool is optimally configured for cutting the project, you may want to place Commands that set the speeds, cutter size, or other features of ShopBot operation that you will always use with this file:

```
'Setup lines for this file
 VS, 1.25, .45, , 2.75, .80          'to set speeds
 VC, .25           'always using a .25 in cutter for this file
 SA                                   'distances will be absolute
```

TIP: Blank lines and spaces at the start of a line are ignored by the ShopBot software when reading a Part File. This means that you can freely use blank lines and indentation to organize the appearance of your Part Files and make them easier to read and understand.

Using Variables. When creating a Part File, it is sometimes useful to indicate a value as a 'variable' rather than as a fixed number. Using a 'variable' just means that we will use a word to temporarily stand-in for a number. For example, let's say you are making a Part File to cut out a fancy grating for the front of a series of cabinets. Sometimes you will be cutting this grating in 3/8" material and sometimes in 1/2" material. There are dozens of Z plunges in the file and you know you don't want to have to rewrite all these plunge values for different depths of material. One way to deal with this problem is to use a variable for the Z depth for the cut. Variables in ShopBot all use the '&' (ampersand) as the first character and are created like: &Zdepth = -0.52. Creating a variable like this at the start of a file means that we have assigned the value of -0.52 to every instance of the &Zdepth variable that follows in the file. So we would just write our Z plunges like: MZ, &Zdepth. Then if we want to change the plunge depth in a file, even though there may be numerous plunges, we only need to change the one variable at the top of the file.

```
                    'Setup lines for this grating file
&Zdepth = -0.52     'change this variable to change cut depth

JZ, .5              'pull up
J2, 10, 10          'goto position
MZ, &Zdepth         'plunge
M2, 30, 10          'cut slot in grate
JZ, .5
J2, 10, 15          'continue cutting slots …
MZ, &Zdepth
M2, 30, 15
JZ, .5
J2, 10, 20
MZ, &Zdepth
M2, 30, 20
JZ, .5
… and so on
```

This example only scratches the surface of what can be done with variables in a Part File, but it should give you an idea of how variables may be used. Note that there are at least a couple of other ways to deal with the problem of needing to change Z plunge depth throughout a file. One method is to stick with fixed numbers and use the 'search and replace' function of the Editor to quickly make a change to all the Z values. Another method is to change the depth of plunging by altering the Z-scale/proportion (one of the [FP] parameters) at the time the Part File is executed. There are usually many ways to 'skin the cat' in working with Part Files. Have a look at some of the ShopBot sample files for other examples of what can be done in Part Files. Sample files are found in the SBPARTS folder and all start with the name ‘S_’.

 Copyright 2008 ShopBot Tools, Inc

Nesting or Embedding Part Files. One Part File can call up and execute another Part File. You use the [FP] Command in the first file in order to start the second file. When the second file finishes running, action returns to the next line of the first file. This process of starting one file with another is referred to as nesting or embedding files. Typically, you might have a 'master' Part File that calls up and places various components that will be cut from the piece of material currently on your table. This master Part File can position the tool for cutting the components and thus manage the layout of the project. An example of placing a single part in multiple locations is described above in 'Temporary Zero' discussion of the Table Layout Section above and the sample code partially reproduced here. You use the 'offset' function in the [FP] Command to cause the file to be cut from the location that the master file moves the tool to. The parts don't have to be the same. You can just as easily use nesting to place different parts of a project in their correct location.

```
'Start of master Part File ALLPARTS.SBP
   M2, 10, 4                    'move to location to start first part
   FP, MYPART.SBP , , , , , 2  'execute with 2D offset (used commas here)
   M2, 20, 4                    'move to location for start of second part
   FP, MYPART.SBP , , , , , 2  'execute with 2D offset
   M2, 30, 4                    '... and so on
   FP, MYPART.SBP , , , , , 2
   M2, 40, 4
   FP, MYPART.SBP , , , , , 2
```

The process of nesting is an example of what in programming would be called a sub-routine or sub-program. Sub-routines are useful for organization and efficiency, and more generally provide a way to 'structure' your Part File work. As an example from our shop, we have a drilling/countersinking routine that we use for many different purposes including making the table-tops for ShopBots. We save the counter-sinking routine for just one hole as a Part File. Then we call it as a sub routine within any file that we need to drill and countersink.

Here's another example of embedding files. Let's say that you have a file that you would like to cut with multiple passes. The [FP] has a 'repeat' parameter for multi-pass cutting but this function is only appropriate if the file is a 'template file' with no Z moves inside the file itself – in this case you specify with [FP] how deep each pass will be made. If your file already has Z movement in it, you can use file embedding to create multi-pass cutting.

```
'Start of master Part File MULTPASS.SBP
'-assume that your file was initially set up to cut a ¾" part in one pass
'-so we want to plunge about .25 or 1/3 of the depth each of 3 passes
   J2, 6, 12                              'move to location for part
   FP, MYPART.SBP , , , 0.33 , , 2, , 1  'first 1/3, 2D offset, from zero
   J2, 6, 12                              'back at start
   FP, MYPART.SBP , , , 0.67 , , 2, , 1  'second 1/3, 2D offset, from zero
   J2, 6,12                               'back at start
   FP, MYPART.SBP , , , 1, , 2, , 1, 1 'last 1/3 … add TABBING to hold
```

Creating Part Files from other Software

Most of the time the cutting you will want to do will be too complex to 'write' the toolpath movement yourself. Fancy curves or lettering, although being exactly the same in principle as the files described above, may require tens of thousands of movement instructions. It's not practical to do it by hand -- this is what computers and design software are for. Thus, for most of your work you will use other software to create the ShopBot Part File for your project. This creation of complex Part Files can be a 1-step or 2-step process. It is 1-step if you have software that directly generates ShopBot Part File instructions. It is 2-step if your

 Copyright 2008 ShopBot Tools, Inc

software generates an output file that is then converted by the ShopBot software to the Part File format.

One-Step to a Part File. With the ShopBot PartWorks design software, you can set the size of your material and the 0,0,0 location, design a project, create the toolpaths to allow for cutter waste and order of cut, and generate the ShopBot Part File (.sbp). A number of software packages for signmaking, computer-aided machining (CAM), and cabinet making also generate direct ShopBot Part File output (ArtCAM, ProfileLab, CabinetWare, VectorCAD/CAM, MasterCAM and others). In this case, you will just use your software package to design your project, save the Part File, and run it on your ShopBot tool from the ShopBot Control Panel.

Your ShopBot will also work with the G-code format CNC files that are output by some older CAD/CAM software programs. [See later section for details.]

Two-Steps to a Part File. Other CAD or drawing software packages will generate output in a format that can be converted to a Part File before it is run. The most common of these is the .DXF format output that can be produced by virtually all CAD programs as well as by drawing programs like Corel Draw. In this case, you save the work in .DXF format in your drawing program, start your ShopBot software, and then use the [FC] Command to create a Part File. After this, you are ready to run the file. You can also bring your file into PartWorks to set the toolpaths and generate the ShopBot file.

In addition to .DXF, the ShopBot converter will translate 'hpgl' format files, and it will create 3D cutting files from grayscale bitmap files (.bmp & .jpg). There is also a converter to go from a ShopBot sbp file format to a dxf file. See the Working with Converters section below for details on using these built-in converters. You will also find the section on 'Getting from your Ideas to ShopBot Products' of interest.

TIP: Even when you have used other Software to create a Part File, it can be useful to do a little additional editing of the file yourself. You needn't mess with the details of the cutting -- they can be quite extensive. But you may want to consider adding documentation and setup information at the start of the file and perhaps introducing PAUSES at critical locations.

Copyright 2008 ShopBot Tools, Inc

From Your Ideas to a ShopBot Part File

A Personal Robotic Tool can turn your ideas into realities -- or into thousands of copies of those realities. With the revolution in affordable robotic tools started by ShopBot, you now have this powerful technology ready to put to work in your shop. As the revolution spreads, the process of getting from idea to product will become easier and more efficient. But, even though the eventual potential is amazing beyond imagination, today there are practical hurdles in making the transition from your concept to the creation. The hurdles are manageable, but they must be recognized and scaled. Here we briefly take up the things you need to consider in getting your designs, concepts, or ideas into production. What this all amounts to is selecting the approach to creativity and productivity with a robotic tool that is appropriate to your own particular needs, skills, and tasks ...in short, figuring out how you are going to make Part Files.

Representation to Realization

By now you probably recognize that what you need to produce a product or creation with your ShopBot is a Part File. The Part File is the computer 'realization' of the detailed instructions for making your project or product. While we still have to deal with holding it down and cutting or machining it, these practicalities are pretty straightforward and easy to understand, at least in principle. But the big issue, and the hard part about personal robotics, is how do we get a Part File 'realization' from an idea, concept, or 'representation' that is in our head or sketched on a napkin?

Copyright 2008 ShopBot Tools, Inc

Getting from your representation to the realization is getting easier as software developers continue to improve their products and make them more user friendly. Here we'll point you to some of the many approaches you can take – some are easier than others. Generally, using a CAD/CAM software is the easiest way to go but there may be reasons that you prefer to use one of the methods described below, so we'll include them here.

We will outline here the possible approaches to product realization in terms of the different ways of representing your design or concept ... and then consider what is involved in producing the Part File 'realization' using that form of 'representation.' The fundamental point is that there are many different ways to represent your idea -- from pencil and graph paper and a little geometry to a sophisticated (and expensive) 3D Cad/CAM/surfacing program -- and given a particular form of representation, there are usually multiple ways to get to the product. This diversity means that there are choices that have to be made, but it is means that you can select an approach that you are comfortable with and that is suited to your style, using this approach as your way of 'working into' robotic production and creativity.

Representation considerations -- a non-exhaustive list from pencil to programming -- Here's how you can do it:

You Can Draw it on graph paper with a pencil and then type it in

As we have discussed above in the context of describing Part Files, for a simple machining project it is possible to sketch out what you want to do on a piece of graph paper and then type in the Commands needed to carry out the moves. You'd use the graph paper so that you could figure out the coordinates for the moves or locations for circles or rectangles. Then you would translate to ShopBot geometry, using the preview screen to check how well you've done ... along with a little bit of trial and error. The [CP] Command, which is used to place a circle at a known center point, can be quite useful with this approach, for example in laying out a drilling or machining pattern that uses circles or arcs. This straightforward approach of writing a Part File by hand is also useful for projects that involve multiple repetitive drilling of arrays of holes or placement of small part (see the S_ARRAY.SBP sample file). Hand written files can be somewhat complex too. Have a look at the S_SBLOGO.SBP file which creates the ShopBot logo -- Gordon Bergfors created this one on a piece of graph paper, but you probably would not want to try anything much more complex than this by hand.

Who for: Creating a Part File by hand is a technique that will probably be of use to all ShopBotters for some types of projects. It's also a good way to learn about how a Part File works ... and to add features to a file that is generated by other software.

ISSUES - Not too hard for simple projects once you get the basic idea of how your tool works ... but this grows increasingly tough with a large project or where any type of curves are involved.

You Can Use ShopBot's PartWorks (a 2-D CAD/CAM Design program)

Included in the purchase of your ShopBot is PartWorks, a design program that allows you to draw and design, import dxf and other files, set toolpaths and generate Part Files. With PartWorks, you can V-carve, create raised or recessed objects and letters (area clear or pocket), machine along vector lines or set toolpaths to compensate for cutter waste. The PartWorks interface is very user friendly allowing for a shorter learning curve when compared to many other CAD/CAM programs. This is the recommended approach for most ShopBotters ...

 Copyright 2008 ShopBot Tools, Inc

Who for: If you already think in plans and dimensioning of objects, this is where you should start. Even if you don't the PartWorks CAD/CAM software is pretty straightforward to use.

ISSUES - The PartWorks CAD/CAM software may be all that you need for your type of work. But if you are interested in 3D modeling you will need additional design software. PartWorks 3D can toolpath it for you, but does not provide a true 3D design environment.

NOTE: Know where your software package places its point of origin. When you bring the design into ShopBot Control software directly as a .dxf file, you may need to relocate the 0,0,0 location on the Bot so that it matches or is registered to the 0,0,0 in the design file. Or, you can bring the .dxf file into PartWorks and position it correctly on your material.

The Limitations of CAD without CAM. The fundamental issue with CAD programs alone, and with drawing programs such as Corel Draw, is that they are not intrinsically oriented to producing tool paths. This creates two specific problems. First, when CAD programs create the DXF file that ShopBot will use to build a Part File using the .DXF converter, the entities in it are usually exported in the order that they are drawn or edited -- and this may not be the order that you want to cut them in. Thus you may need to create an initial draft or underlying drawing of your part, and then trace the final cutting path around it, saving the latter as your ShopBot-ready DXF file in order to get the cutting ordered as you want it to occur. The second problem with CAD and drawing programs is that they are not aware that you will be using a cutting tool that has a diameter. Thus CAD and drawing programs cannot automatically compensate for the diameter of your cutter and shift the path to the outside or inside of the part. This is where PartWorks' CAM function comes in. PartWorks uses the tool database to calculate the correct offset automatically.

Raster to Vector in PartWorks, Digitizing in 2-D
We are frequently asked if there is an easy way to get from the known outline of a part or pattern, to cutting it in ShopBot. This might be a piece of (bit-map) clip-art or a scanned image of the outline of a part. There is no perfect method for this but you can make it work with a little fussing.

1. A first approach is to do a 'raster-to-vector' conversion of your clip-art or scanned image in PartWorks. These images are initially bitmaps (or rasters), being made up of thousands of little computer screen pixels. The PartWorks program will do it's best to help you turn this image into lines that can be cut. A conversion process generates a vector or 'line' version of the file. In a perfect world the conversion would give you a smooth and precise cutting path. However, you will usually need to do a little editing of these paths.

2. You might also consider 'probing' your part with a digitizing routine in order to get the data for producing an outline file. There is a virtual tool in the ShopBot Control Software called Copy Machine that you can use in conjunction with a digitizing probe to produce either a cutting file or a file that you can import into a CAD/CAM program for further processing. This process can be made to work with a ShopBot Probe, an electrical contact stylus, or an edge sensing optical scanner.

Who for: At this point, any of these approaches is going to involve some puttering on your part. They can be made to work, but can each be a little challenging until you work out the specifics for your own needs.

ISSUES - As personal robotics tools become more popular, there will be an increasing amount of affordable software available to automate the input of existing drawings and parts. We expect a lot of improvement in these specific areas.

 Copyright 2008 ShopBot Tools, Inc

ShopBot Can Help with 3-D Representation-Digitizing in 3-D

The reason that working in 3-D is difficult is because it is hard to generate the locations of 3-D points and entities in a visual 2-D space. One method to work around this difficult is to use a physical model of the object that you want to realize in your Part File, collect the 3-D information that defines it, and then use this information for the generation and modification of the Part File. One very straightforward approach described in the Accessories section is a 3-D digitizer. Your ShopBot can use such a device to sample the 3-D surface of an existing part to generate location data for use in other software to directly create a cutting file. The basic idea is to systematically sample a surface by making a series of Z-Axis plunges. A plunge is made until contact with the surface occurs. The location data is recorded for the contact point and the probe is pulled up, moved to the next location, and the next plunge is made. The contact is sensed by a switch closure or other electrical event. The probe can be as simple as a stylus attached to a micro switch, or, in the case of a metal part, just the electrical contact of a stylus with the part. You can use the Copy Machine under the tools menu of the ShopBot Control Software to set up the probing process. Using this virtual tool in conjunction with a probe allows to you generate a toolpath by simply probing an existing part or object.

Who for: If you are interested in 3-D modeling and/or you already have sample shapes you would like to duplicate, this is an approach you should explore.

You Can Use 3-D CAD/CAM

The ultimate software tool for getting from your idea to creation of 3-D machining instructions are 3-D CAD/CAM/Surfacing/Modeling programs (CAM= Computer Aided Machining or Manufacturing). These tend to be high-end programs (>$2000) and require a substantial amount of training and experience to become proficient. In theory they are capable of producing CNC-oriented Part Files for practically any type of project. The work can be challenging, but this is how CNC parts are programmed in engineering and industrial settings.

Other 3D CAD/CAM programs are oriented to artistic type of relief carving. For example, ArtCAM Pro is a user-friendly package that will allow you to import a bitmap, create 3D cutting paths and generate sbp Part Files.

An additional 3D alternative that many ShopBotters have found helpful is to use an intermediate priced modeling/surfacing program, such as ‘Rhino3D’ to create the representation of their part. This model is then processed through the PartWorks 3D CAM program to create the Part File. This approach gives a relatively easy to use modeling environment and the quick and easy generation of a cutting file.

Who for: If you are doing detailed 3-D machining you may have no other practical option than a CAD/CAM program. But keep in mind that there will be a substantial learning period getting used to working in the 3-D software, and that any particular part will require significant design time. Don't start here if you are a computer novice.

ISSUES - Once the price, time, and difficulty issues are behind you, there are few limitations with 3-D CAD/CAM/surfacing software. This approach is the way that industrial CNC is usually driven.

You Can Use Special Purpose Software

Several industries already have specialized, well developed robotics software available that is oriented to creating cutting paths for specific kinds of products or objects. Sign-making

Copyright 2008 ShopBot Tools, Inc

software, cabinet software, boat-design software, are good examples. While somewhat expensive, some of this software offers an easy and efficient way to create a Part File -- often virtually automatically. It can thus be well worth the cost.

Who for: If appropriate specialty software is available for your business, you will want to consider it. With effort, just about anything can be done with the approaches described above, but a special purpose software may get you going in a hurry and be cost effective ... some of it is extremely capable.

ISSUES - The only significant issue with a specialty program is whether it will generate a .SBP or G-code file that your ShopBot software can easily run. If in doubt, test a sample output file from the program with the ShopBot Software (always available for download from out web site).

A First Project: Surfacing Your Table

One of the first things you'll want to do after getting your ShopBot together and running is smooth your work surface. Surfacing is actually your first project. In case you didn't follow these instructions in the assembly manual, here are a set of general procedures for surfacing your table.

By this point you should have the table surface material mounted on your ShopBot. You should have attached the support board to the table with the carriage bolts (recessed) included in the table hardware. Now, attach a sacrificial board to the underlying support board by countersinking 1 ¼" sheetrock screws in it and screwing them into the support board. If there are warps in the sacrificial board, check to be sure that all the sheetrock screws are tight.

You will want to make sure that you followed the instructions to square up your Z axis before you start. The smoothness of the surfacing you do will depend on the Z axis being perpendicular to the table. However, don't be too disappointed if your surface has distinct tooling marks. You can't really get the Z axis perpendicular until you've got the table flat once. After you've finished this surfacing, you will actually be able to adjust the Z axis with even more precision because you will have a tool-defined, flat plane to measure against. But to keep the tooling marks and striations to a minimum during the initial surfacing ... get the Z as straight as you can to begin with. The larger the diameter of the cutter is that you use for surfacing, the less time it will take. But a wider cutter makes it more likely that you will have tooling ridges from the process and increases the load on your router. A .75"-1.25" diameter bit will probably work well. Always enter the cutter diameter for a new bit in your software using the [VC] Command. For the instructions below, make sure you are in Absolute Move Mode [SA].

This little project is going to generate a lot of sawdust. We'd recommend you get your dust collector hooked up and running before starting the actual cutting.

The ShopBot Control Software has a Virtual Tool called Tabletop Surfacer that simplifies the process of flattening your table. When you launch the Tabletop Surfacer, input your table dimensions, bit diameter, percent of bit overlap, the corner that you will be starting at and the direction that you want the tool to move. Click on the *Make my surfacing file* button and you will be prompted to type in a name for your cutting file. The file will be saved in the SbParts folder so that you can call it up with the FP command whenever you need to surface the table in the future. A prompt will ask if your bit is in the correct starting position and answering Y to this prompt will run the surfacing file that you just created.

 Copyright 2008 ShopBot Tools, Inc

NOTE: You may want to try this up in the air first before you really cut the surface, (you can do a few passes to make sure all is going right, then stop the action with the space bar, jog home [JH], and start over … you can also try this in Preview Mode first)

If you have a very uneven surface and need to take off a lot of material, you may need to do it in several passes to keep the router from bogging down.

NOTE: Starting corners for the [CR] Command that is used in the Tabletop Surfacer are set up like this. Looking down from the top, the corners of the table are numbered clockwise, starting in the upper left corner. So we are starting at 0,0 which is position 4. Just trust on this …

Upper Left = 1 Upper right = 2

Lower Left = 4 Lower right = 3

Copyright 2008 ShopBot Tools, Inc

Working with Your Computer

USB Connection

Your PRSalpha uses a USB communication link between your control box and computer. The first time ShopBot Software runs on your computer it will attempt to automatically connect to your ShopBot. If it does not find the ShopBot Control Box, you will get a yellow pop-up box that gives you several options, one of which will step you through identifying the correct connection port. The software will then remember the correct setting.

There may be an occasional situation when the software is having trouble identifying the right port where it is necessary to explicitly tell the software that you are using a specific port. To do this, start the software. Answer the connection failure prompt with "C" to Continue without a connection. Then use the [VI] Command to set the Port number to the one you believe is correct. The software will check again to see if a connection can be established at that port number. See the Troubleshooting section for further suggestions on problems getting connected to your tool.

Organizing Your Computer for ShopBot Projects

As you begin to generate Part Files, you will want to create separate directories/folders for the Part Files related to various projects or tasks. We recommend creating a 'SbParts' folder in your C:\folder (this is actually done automatically when you install the software). Once you have a SbParts folder you can make sub folders for more specific projects.

Special Purpose ShopBot Files

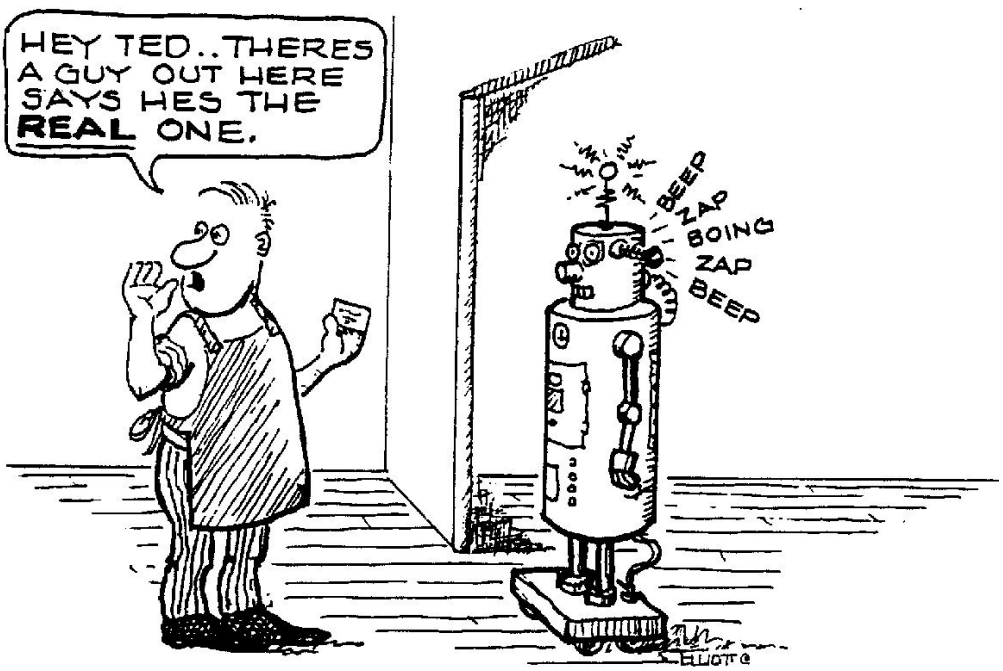
There are several special purpose ShopBot Files. These files are all basically regular Part Files though they have different file extensions that identify their function. These files are all 'text' files that you can read (or modify) with a text editor. Because they are written in Parts File format they can also be executed with the [FP] Command.

The special file type that you are most likely to encounter is the SHOPBOT.INI file. All of the current Settings and Values of your tool are being continuously saved to the hard disk on your computer. This means that whenever you make changes to the settings, these changes will be preserved when you restart your ShopBot the next time. Settings are saved in a file called SHOPBOT.INI. This file is read whenever your software is started. It is automatically saved after every Command you enter. If you would like to save current Settings and Values for re-use in a particular situation or with a particular Part File, you can use the [US] Utility to save the current Settings and Values to a File that you specify. Then run this file with [FP] to re-establish the saved Settings and Values any time that you need them. This is easiest if you name the file with a .SBP extension.

When the SHOPBOT.INI file is first created its parameters are derived from the default Settings & Values file for your tool. This file is the .sbd file that you are prompted to select the first time you run the software. You can modify this file, but the idea is that this file will always contain the most basic starting settings for your ShopBot – you may only want to make sure it has the right Unit Values

If you find a need to have any special settings loaded at start-up or a particular file automatically started, you can put all those instructions in a file named SHOPBOT.OPT. This 'option' start file will automatically be read and executed right after the SHOPBOT.INI file is processed at start-up. The format for this option file is that of a standard Part File. Place it in your ShopBot directory. As an example, if you wanted your ShopBot to always start with the arrow-key-movement display, create a file with the line, SK , and save it as the SHOPBOT.OPT file in your ShopBot directory (C:\Program Files\ShopBot\ShopBot 3) Remember the arrow-key functions only work if you are attached to a tool.

Copyright 2008 ShopBot Tools, Inc



Working with Your Tool

ShopBots are used for a variety of purposes and in a variety of settings. For this reason, it is difficult to anticipate exactly how a tool should be configured for a particular application. In this section we review some of the general issues you will want to consider in setting up your tool for your specific projects. Keep in mind that the best information for your application will come from your previous experience working with your materials and cutting or machining characteristics and from doing some test cutting with different software and mechanical configurations of your tool.

Many of the same rules that apply to using a router by hand also work well when using a CNC machine. However, the advantage of programming the perfect cutting speed and maintaining that speed throughout a series of complex cuts puts you at a whole new level of accuracy and repeatability. For those of you experienced with routing by hand or mounting one under a table, many of these techniques will be second nature, but for those of you just starting out, you will need to consider the following variables very carefully.

Speeds, Feed-Rates and Rapid-Positioning

Before we even begin to consider what speed you want to be cutting at, recognize that there are limits to how fast your ShopBot will go. If you are not familiar with stepper motors, this will probably seem counter-intuitive but as stepper motors go faster, they lose power; the slower they go, the more power they have. Spindle equipped alpha tools can cut at 6 to 10 in/sec and jog at about 30 in/sec.

ShopBot has two Speed modes: One that is called 'Move' Speed and one that is called 'Jog' Speed. All movement is done either as 'M' Commands (Move Speed) or 'J'og Commands (Jog Speed). The movement in each case is equivalent (similar resolution and vectoring), but each Speed mode has a pre-assigned speed (Speeds are set with [MS], [JS], or [VS]). Move Speed is the speed that you've assigned for making movements that involve cutting or machining. Move Speed is the rate your material sees the cutter at and is thus your 'Feed

 Copyright 2008 ShopBot Tools, Inc

Rate.' All ShopBot moves are fully 'speed vectored' which means that the cutter moves at the same continuous speed whether it is moving in the X axis the Y axis, on a diagonal, or in a circle or curve (unless speed is being 'ramped' up or down at the beginning or end of a movement, see below). The cutter velocity is constant.

Jog Speed is the speed designation for the rate at which the tool is moved when it is not cutting or machining material. Jog speed is normally set faster than Move Speed and Jogs are thus the method for the rapid positioning of your tool.

Choosing the correct Move Speed for your project is one of the most important choices in getting a good quality and efficient cut from your tool. There are a number of issues and considerations here. While we can offer some guidelines, there is nothing more important than your own experience with the cutter and material in making the decisions about cutting speeds. ShopBots working at 2"/sec or less have ample power to break bits and stall routers ... thus at Move Speeds the power of your ShopBot is not likely to be nearly as important an issue as your cutter type and size, the material you are using, the cutting direction, and speed and power of your router or spindle

At the two extremes in setting feed rates, you could be moving slow enough to burn or melt your material, or fast enough to give you poor edge quality, cause chatter, bog down your router and break bits. Somewhere between the two will be a move speed that has plenty of power to cut at a fast rate without sacrificing edge quality. For ¾" hardwoods, that might be between .75" and 1.25" per second and for ¾" plywood you might be traveling between 1" and 1.75" per second. For thicker or harder materials, a second pass may be in order. Here is where some trial cutting and close inspection will really help. Whatever your optimum speed turns out to be we recommend that you program that speed into the top of any file that you will be using repeatedly, so that you always have a reliable feed rate.

Using Chip Load Calculations to Improve Cutting

One of the challenges in getting good CNC cuts is in selecting the best cutting speed (feed rate) and router/spindle RPM. Selection can be facilitated using the guidance provided by manufacturers regarding the best 'chip load' for a particular cutter. Since chip load (the size of the chunk of material taken by a tooth of the cutter) reflects the combination of how fast the cutter is moving forward into the material and how fast it is turning (Chipload = Feedrate / [RPM x number of flutes]), it gives you starting point speed values for testing to determine the most suitable parameters for any cutting situation. The ShopBot Chip Load Calculator simply provides a quick way to explore these values, and is most useful if you know the approximate desired chip load for a particular cutter and material.
The following are some very rough chip load ranges for cutting various materials. Keep in mind that the most effective chip load depends on the specific geometry of the cutter, so you will want to check with the maker of your specific bit, and then keep your own records on what values work best for you.

Copyright 2008 ShopBot Tools, Inc

The following chip load values were abstracted from the Onsrud Cutter Catalog which is an excellent source of information relating to cutting and machining. http://www.onsrud.com

CHIPLOAD	¼ inch diameter	½ inch diameter
Softwood	.002-.020	.003-.030
Hardwood	.005-.020	.005-.030
MDF	.005-.020	.005-.030
Plywood	.005-.020	.005-.030
Laminated Chipboard	.008-.020	.010-.030
Soft Plastic	.004-.012	.006-.014
Hard Plastic	.006-.012	.008-.016
Aluminum	.003-.010	.004-.012

If depth of cut = 2x diameter then reduce chip load by 25%.
If depth of cut = 3x diameter then reduce chip load by 50%.
Note that there is considerable overlap in the range of parameters and this indicates that you will need to test for the best speed selection for a particular cutting or machining operation. Here's the strategy that Onsrud suggests:

1. Start off using an RPM derived for the chip load for the material you are cutting.
2. Increase the cutting speed (feed rate) until the quality of the part's finish starts to decrease or the part is starting to move. Then decrease speed by 10%.
3. Next decrease RPM until finish deteriorates again, and then bring it back up until acceptable.
4. This optimizes RPM and speed so that you are taking the largest possible chips.

A Suggested Feed Rate Starting Point. For working in most woods, we suggest you start testing at 1.7" per second for your XY Move Speed. Evaluate your bit and material to see if you can work at faster speeds and still get smooth cuts. For some tasks, it may be necessary to move more slowly. Set the Ramp Speed to .4. Ramp Speeds are not of real importance with the relatively slow movement speed at this cutting rate. But as you start moving faster, ramping can assure that you have a smooth start and stop.

A good starting Move Speed for your Z plunge will be about .5"/sec. This should give you plenty of drilling power. Set the Z Ramp Speed at .4.

TIP: It is hard to over-emphasize the importance of a sharp bit for smooth, clean cuts. Bits will seem to wear quickly because it is very easy to accumulate mileage with CNC cutting. Do not be tempted to use an old dull bit for reasons of economy. The cut will be rough and the cutter will exert extreme forces on your tool.

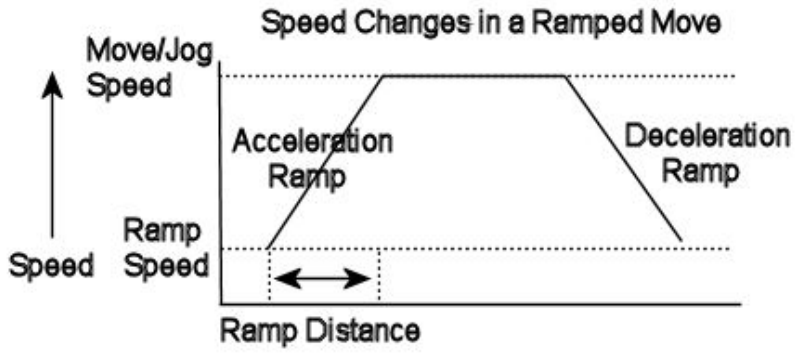
Suggested Starting Jog Speeds. Jog Speeds should be set to speed up your program run time by moving you to the next cut as quickly and efficiently as possible. Please note here that an efficient tool path can be just as important as a fast tool. The danger of going too fast with stepper motors is that there is an inverse relationship between speed and power. If you set your speed too high, your power will drop and you can risk stalling the motors and losing position.

Copyright 2008 ShopBot Tools, Inc

Acceleration and Deceleration 'Ramping'

Acceleration ramps and deceleration ramps are the gradual increases or decreases in movement speed inserted at the beginning and end of a move segment or series of segments. We usually just refer to this action as 'ramping'. Ramping helps your tool move smoothly when executing moves, particularly at faster movement speeds. Without ramping, your tool would 'bang' around and generate excess vibration as it changed direction between moves, or at stops and starts. Excess vibration might show up as roughness in the quality of your cut, or the strong vibration or jolts between rapid moves might be violent enough to cause your tool to loose track of its location. If you are just starting off with your ShopBot, have it make a couple of Moves and Jogs. Listen carefully and you will here the ramps at the beginning and ending of the movements. The capability to ramp is built into the software. In most cases the default settings will automatically produce ramping for you that is appropriate for what you are doing. But you do have full control over ramping and this section will fill you in on the principles. These are relatively 'fine points' of CNC work ... if you're just getting started, skip this section for the time being.

Ramping for your ShopBot is linear. In an acceleration ramp your tool starts moving at the designated Ramp Speed and increases speed linearly until it arrives at full Move Speed at the end of the Ramp Distance. Similarly, in a deceleration ramp at the end of a move segment, your tool decreases speed linearly from its Move Speed to its Ramp Speed over the Ramp Distance. If desired, you can exercise full control over these characteristics of your tool's behavior as you can define the Move Speed or Jog Speed (using [MS], [JS], or [VS]) of your tool and you can also set the starting/ending Ramp Speed and the size of the Ramp Distance (both using [VR]).



Ramping in the flow of movement segments. The above description of ramping covers the basics, but how ramping works is a bit more complicated -- that's because sometimes, your tool should ramp between movement segments, and sometimes it should not. Consider cutting out a rectangle: At the start and end of each side or movement segment in the rectangle your ShopBot should ramp-up or ramp-down so that the direction change at each corner does not result in an abrupt change in the action of the tool -- one that might create unwanted vibration which could show up in the cut. In this case, every segment should be ramped. If you 'air-cut' a rectangle using the built-in rectangle function [CR] you will hear the ramping at the beginning and end of each leg (move segment) of the rectangle.

Copyright 2008 ShopBot Tools, Inc

On the other hand, consider cutting a smooth curve: Such a shape might be composed of hundreds or thousands of small movement segments that flow naturally into each other. Here we would want to ramp-up at the start of the curve, cut smoothly at full speed through all the segments of the curve, then ramp-down at the end. We would not want to punctuate each movement segment with ramps. This would be inefficient because it would slow the tool below the speed it could cut at and, because the speed of the tool would be constantly changing, the quality or smoothness of the cut might suffer. Listen to the movement of the tool through a circle using the built-in circle function [CC] to make say a 6" circle. (That is, assuming you are not at the edge of the table, just type: CC, then a 6 for the diameter in the Fill-In Sheet, then hit ENTER twice and your tool should execute a circle in the air.) Even though the circle is made up of hundreds of small segments, there will only be a ramp at the start and end of the circle.

TIP: Notic**e** that in addition to the ramps at the start and end of the circle, the speed of the motors change as the tool moves through the circle. This is as it should be. The changing motor speed occurs in order to maintain a constant vectored speed of the cutter as the movement makes transitions between being predominantly one axis and then the other.

Finally, consider the situation of a typical Part File: Real files will normally be interspersed with straight segments having abrupt changes in direction which need ramping and other segments that are small parts of gradual curves and which should not be ramped (for example, in letters like a 'B'). Your ShopBot software 'looks ahead' at a block of move segments, computing for each segment whether a ramp is needed. If the change in speed is too abrupt between move segments -- sharp corner rather than a smooth curve -- your tool will ramp down before the angled corner and then ramp back up to speed after it. If an angle transition between segments is slight, then your ShopBot maintains constant speed and moves smoothly through it. The way the software figures this out is as follows: For each axis the change of speed for each move segment is determined. If the change in speed is greater than the ramp speed, then the tool is set to ramp into and out of that corner. The sensitivity of this threshold can be further adjusted with the 7th parameter of the [VR] Command. At 100% (the default) the software uses the ramp speed for the axis as the threshold. Adjusting the percentage up increases the speed change that must occur for a ramp to be set, adjusting down decreases the threshold speed and thus makes ramping more likely. The final parameter of this command, which you are unlikely to need, sets the minimum size of segment (in steps) that is required for a ramp test to be made.

Now for a few last details you may or may not want to know about. Imagine a cutting file for some detailed lettering in a small sign. Frequently in such files (e.g. generated by a sign making or lettering program), there will be portions of letters that are created by hundreds of small back and forth movements of the cutter point. These segments may all be at angles that require ramping, but the distances may be too short for the ramp to be completed and the tool to get back up to speed. If we didn't apply an alternate strategy, you would get a saw tooth speed profile that would produce a lot of vibration in the cut. In such cases, the tool movement speed will default to the ramp starting-speed and the tool will continue moving at this constant reduced speed until it gets to a segment that is long enough to actually ramp up to full speed. Built-in circles and arcs (i.e., CC,CG,CA) handle the small ramped segment similarly. In a large circle or arc, there will only be a ramp at the beginning or end. In a small circle where the angle is changing quickly, rather than having a lot of ramps between little segments, the circle is executed with a continuous move speed equal to the ramp starting speed. For circles and arcs, the diameter at which the circle is executed at the slower speed can be set using the 8th parameter of [VU]. The default setting is .20, meaning that circles or arcs with diameters less than .2" (true path) will be cut at ramp speed. If you wanted to change this so that circles less than .35" would be cut slowly, then you would enter [VU] followed by: , , , , , , , .35".

 Copyright 2008 ShopBot Tools, Inc

If you are interested in ramping issues, you will also be interested in reading about Movement Blocks (see the Part File Programming Handbook) and how they are used and defined. Working with blocks can give you some additional control over the smoothness of a particular cutting file.

TIP: Even with ramping providing the smooth start and stop to tool movement, some cuts will require 'lead-ins' -- movement into the cut from outside it. In particular, lead-ins may be required where the bit needs to be plunged into the material before a part is cut out. This is because at the point the router starts to move, the direction of the forces pulling it into the cut will change from those that are exerted during plunging. These forces matter because there is a small amount of run-out in the shaft of the router and a small amount of flex in your ShopBot and in the presence of side force these run-outs can result in a tooling mark. Careful setup of your tool will minimize this flex and these types of problems, but lead-ins may still be required in some situations. These can be programmed into the cutting file and arranged to occur in waste portions of your material.

Direction of Cut and Tool Path

Optimizing your tool path and direction can also improve the quality of your cut. When using a router by hand it is prudent for safety reasons to go against the direction of rotation (conventional cutting), however that is not always the best direction to travel for a smooth edge. Going in the same direction as the rotation (climb cutting) can produce a cleaner cut, especially when doing a finish pass using straight bits. Keeping this in mind, if I was cutting an oval picture frame and the bit was rotating counter clockwise, I would want to travel clockwise on the inside of the frame and counter clockwise on the outside.

Tooling

The type and quality of your bit is another important decision you will make in efficiently producing high-quality products. Although cost is always an issue in production, the least expensive bits are usually not the most cost effective. To fully evaluate the worth of a bit, you must be able to calculate how long the bit will last (in linear footage or number of parts made) before re-sharpening and if it's worth it to re-sharpen the bit at all. The other important issue here is the quality of the cut. It does not save money if you have to rework all your edges after cutting. So, you can see there is a little bit of experimentation to be done to find the ideal bit.

Here are just a few suggestions to get you started in your search:

- ¼" is a good diameter for cutting sheet stock like ¾" plywood or MDF and thinner stock in acrylic or aluminum. Any smaller than this and your chances of breaking a bit increases.
- Solid carbide lasts much longer than high-speed steel, but is also more expensive.
- Although carbide tipped straight bits are an excellent value and easy to re-sharpen, a spiral end mill will produce a much smoother edge and maintain it at a faster cutting speed.
- A down spiral will help keep thin stock on your table, while an up spiral is better for thicker materials because it removes the shavings fast, which will keep the bit cooler and make your dust collection more efficient.
- There is also an up and down spiral bit for when you need a clean edge on both the top and bottom of materials prone to splintering.
- Whenever you can, buy ½" shank bits because they are less likely to break or deflect in a cut.
- There are now special CNC bits designed to hog out parts fast; the cut can then be followed by a different bit making a clean out pass.

Copyright 2008 ShopBot Tools, Inc

ALERT!!! Sharp Cutters Make All the Difference!

Clamping Your Material

Now that you have a tool that will cut material in exactly the manner that you want, you will find that your next major challenge is determining how to hold your part in place while it is being cut or machined. Unfortunately, every project or production situation is different and there is no single best answer to the hold-down question. Here we describe some of the techniques you will want to consider. In the same way that how you jig-up for any project can make a big difference in your efficiency and the quality of your work, your creativity in clamping your CNC work will have a similar importance. On our web site there is a ShopBotter Forum that has an ongoing discussion of techniques for holding work pieces in place. You may find this ongoing discussion of interest or be able to get specific suggestions for your own project.

Clamping Concerns:

- One of the biggest problems is holding small parts that have been cut loose from their substrate.
- One of the biggest dangers is hitting a clamp or screw with the router bit.
- Thoughtful planning is required here. More importantly, keep in mind that your highest priority is that the part be held absolutely rigidly. It is not possible to get a smooth accurate cut if your part moves when the router is machining it.

Temporary attachments. Screws, nails, and tape can all be effectively used in the right situation for attaching material to your table for cutting or machining. These are probably best where you are only doing a few runs of the cutting. The greatest danger is hitting a screw with the bit. Making sure the clamping screws are well out of the cutting path is the best way to avoid this. A trick is to have your ShopBot use a special Part File to put some starter holes for screws in appropriate locations before you start the cutting operation. This way you can make sure they are in all the needed areas but also clear of the cutting path. If you think you might be doing a lot of this type of clamping, consider using an air-gun with plastic nails. Where the cutting is not too aggressive, double stick carpet tape can provide an effective hold-down. It can be particularly helpful with small parts.

Clamps. Some types of projects can be adequately held with clamps (C-clamps, pistol-grip clamps, or spring clamps) or with clamping fixtures mounted on your table. The latter can be general purpose, making use of tracks or slots that you have bolted or machined into your table, or can be specifically attached for the project at hand. With any use of clamps, avoiding hitting them becomes the issue. The cutter path must avoid the clamps, and any positioning moves must have high enough clearance not to hit the clamps, or move around them. There will be times when a few strategically placed holes in your table can be useful for getting C-clamps into clamping position. Plastic 'spring clamps' have an appeal because they will not break your bit if you accidentally hit them. However, be very careful with spring clamps and pistol-grip clamps to make sure they are actually rigidly holding the material in place ... they have a tendency to move.

Vacuum hold down. One popular solution to holding material in position is using vacuum clamping. A manifold of air channels is cut into the work table of your tool, appropriate seals are positioned (or an intermediate, sacrificial, vacuum-penetrating board used), and a vacuum pump is attached to this system. ShopBotters have been successful with a variety of such system. Pumps can vary from a Shop Vac that you carefully control so as not to overheat it, to expensive high-pressure vacuum pumps. There are two basic issues here. The first issue is that if you have a general purpose manifold system, you cannot cut

 Copyright 2008 ShopBot Tools, Inc

through it to cut your parts out without breaking the vacuum and potentially damaging the seals. The alternative is designing a vacuum system for a specific part or set of parts in a project so that the cutting path does not interfere with any of the vacuum sectors -- this solution works well for a production run. The second basic issue is that the effectiveness of vacuum clamping depends on the surface area of the part. Small parts usually cannot be successfully held with vacuum.

Webs, tabs, tool path planning. For some projects it is often possible to hold parts in material by engineering attachments in the pattern or method of cutting the material. For example, leaving a very thin layer of material to hold a part in position can be very effective. After the almost-cut-out pass is made, the part can be removed by trimming it from the sheet or running it through a planer to remove the thin attachment. This method can also be used with a general purpose vacuum system to hold the overall sheet in position without disrupting the vacuum. It may also be possible to plan a web of attachment points into the machining of a part. Very small attachment points can be left that are removed manually later, or are removed in a last pass of the tool. If done carefully the part will not move on this last pass because the cutting forces are relatively small since most of the material has been removed. Attachment points can also be removed with drilling plunges, which further reduces the likelihood of movement. In addition to programming attachment points into the Part File, ShopBot also offers an automatic 'tabbing' system that will place small tabs or attachments at a predefined interval around the perimeter of the cut-out pass for a part (see 'Tabbing' under the [FP] Command). Tabbing can be activated at the time you run the Part File. This system is particularly useful in prototyping or producing single parts because no planning is required at the time the part is being designed and the Part File created.

Dust Collection

As you begin to use your ShopBot, one of the first things you will learn is that it is very easy to generate a lot of sawdust. While there will no doubt be times when you will cut a quick part without worrying about the dust, for your health, for tool maintenance, and for shop housekeeping you should try and arrange dust collection for each project you undertake.

The difficulty with prescribing dust collection is that different methods are required for different types of projects: techniques that are appropriate for sheet material may not work well in applications that are primarily 3-D carving; techniques that work well with routing operations in which no parts being cut out may not work when there are loose parts or cut-outs being generated. ShopBots come with a vacuum formed dust skirt that provides both dust collection and also helps to contain larger pieces material waste that could become projectiles. Our design is a good general purpose collection system for working with plywood and similar sheet materials. You may need to engineer other creative solutions for different types of cutting. Consider using either a smaller vacuum head that relies on high pressure immediately around the cutting tool and that moves with the cutting head -- or the opposite strategy of removing a large volume of air from a larger shrouded area in which the cutter moves.

Copyright 2008 ShopBot Tools, Inc

Working with Accessories

Everyone's projects or production needs are different. Our main goal in creating ShopBot was to produce a tool that could be configured, adjusted, or modified in a wide variety of ways in order to effectively and efficiently serve user needs. In this section we discuss some of the primary methods for adding features to your ShopBot. Over the years we have been continuously impressed with the inventive and creative ways ShopBotters have worked with these methods to enhance the capabilities of their tools. Don't be afraid to experiment. We also encourage you to explore the ShopBot Forum on our web site for extensive reports on features and capabilities that ShopBotters have added to their tools.

Input Switches, Limit Switches and STOP Inputs

Your ShopBot has eight 'Inputs' (PRS has 4) that can be used to sense switch closures or other electrical events (Inputs 3 and 4 are specialized, see below). These eight inputs are accessed on the front (backside for PRSs) of the control board in your Control Box. There are two blue screw terminal connectors, one for Inputs one for Outputs. The diagram in Technical Information Section shows the location for each input. The blue connectors can be removed to make them easier to wire, or to disconnect your Control Box.

Technically, the inputs are 5 volt, TTL level circuits which when 'pulled low' are recognized as 'ON' by the ShopBot software. More practically speaking this means that an Input switch is ON/Activated by connecting that input to ground, either to one of the GND terminals on the terminal board, or if your tool is properly grounded (see the Assembly Guide for grounding instructions), to any metal surface on your ShopBot. Thus, to activate an input, attach it to a switch that when closed connects it to ground. The input switches can also be controlled by TTL level voltage inputs from solid state devices.

Sensing Input Switch Activation with your ShopBot Software. You will immediately see the effects of activating/closing an Input Switch on the input section of the ShopBot position window where the number of the input that is activated will turn green indicating that it is ON.

In general you will use the sensing of an Input Switch to change or alter an ongoing activity during the execution of a Part File. You will use the "ON INPUT(#, #) ... " Programming Instruction to act on the switch event and change the ongoing activity in the running Part File. The first part of the Programming Instruction specifies which Input and event type you are dealing with, for example, "ON INPUT(2, 1) ..." means that you will take action when Input Switch 2 becomes Activated/ON -- "ON INPUT(2, 0) ... " means that you will take action when Input Switch 2 goes OFF (is opened). The action that is to be taken is specified in the second part of the instruction. Usually you will use the GOTO or GOSUB instruction to redirect the action in the file. Alternatively, you might have an ‘equate’ instruction that changes the value of a variable. The use of ON INPUT is more fully described in the Part File Programming Handbook. A good example of the use of Input Switches is the S_ZZERO.SBP Part File that gives an example of how to zero you Z axis at the table surface. Also have a look at the S_PROBE.SBP file for an example of using switch input for 3-D digitizing, and the S_HOME.SBP file for the use of switches in re-zeroing the XY home position of your tool.

The status of Inputs can also be ‘read’ as a system variable. For example, if %(52) is 1, then switch 2 is on; if %(53) is 0, then switch 3 is off (see the Programming Handbook for more information on system variables).

Input Switch #2 and #3. Input #2 and #3 are set up to function as a Limit Switches. By default, anytime that Switch 2 or 3 is activated/ON, the action of your tool will stop and

 Copyright 2008 ShopBot Tools, Inc

you will receive a message that you have reached a table limit. You will need to indicate whether you want to proceed or not. Use the [VN] Command to set up the action that you want on these switch inputs.

Input Switch #4. Input #4 is set up to function as a Panic Switch / Stop Button. Anytime that Switch #4 is Activated/ON, the action of your tool will stop, just as if you had hit the Spacebar, and you will receive the same 'Stopped' message. When this switch is activated it will also have the same automatic effect on Output Switch #4 as hitting the Spacebar. Your tool will not function until the Input Switch #4 is released. Note that Switch #4 always functions as a Stop Switch. It cannot be re-assigned for other use. By hooking the switches up in series, you can set up as many Stop Switches on your tool as you would like.

ShopBot's Remote 'Stop' Button

A bright yellow and red Stop Button is provided with your tool. Once it is properly attached, hitting this Remote Stop Button will instantly stop the movement of your tool. Depressing this switch is not the same as hitting the 'Spacebar' on your computer when your tool is in action. The Stop Button will turn stop the stepper motors as well as turning off the router. The Spacebar simply serves as your panic button to interrupt the movement of your tool, but will not interrupt power to the router or spindle. When hitting the space bar you are able to resume operation without resetting the drivers in the control box. The Stop Button comes with a long cable so that the switch can be mounted in a location that is convenient for your set-up.

Please recognize that the presence of a Remote Stop Button does not alter the requirement for operating your ShopBot safely. You should always stand clear of the tool when it is in movement and away from the path of any flying debris, preferably positioning yourself near the computer controlling the operation and to the side of the tool. A Personal Robotic Tool can be a very safe power tool as long as basic common sense safety procedures are followed.

=== PANIC ! PANIC ! PANIC ! ===

When the tool is executing any movement, the SPACEBAR becomes a STOP BUTTON.

>>>> HIT the SPACEBAR or the REMOTE STOP BUTTON and ALL MOTION OF THE TOOL WILL STOP. You will be prompted for what to do next. For a smooth stop where the movement speed is ramped down and the cutter pulls up, use the 'S' key for STOP.

On a PRSalpha, the yellow and red button provided with your ShopBot is a full E-Stop and works by being hooked up to terminals ES1 through ES4, labeled E-Stop on the control board. It shuts down power to the motors and spindle when activated.

Output Switches

Your ShopBot Control System provides eight Output Switches (PRS has 4) that can be activated from the ShopBot Software and used to signal other electronics or activate relays to turn on or off accessories. The Output Switch circuits provide a 5 volt TTL level output (about 5 mA max). The Outputs are accessible on the control board, at the blue terminal blocks at the front of the board marked 'Output' (back of board for PRS). This terminal block also contains a ground and 12 volt dc line (see diagram in the Technical Information section).

Because the Outputs circuits provide relatively low current they can be used to directly drive only a low power device such as an LED or another piece of electronics that is compatible with TTL power levels. To operate a relay for controlling an accessory, you will need to

 Copyright 2008 ShopBot Tools, Inc

create a simple transistor driver circuit for the relay. This requires only a few parts that can be purchased at Radio Shack or another electronics store.

Turning them ON and OFF. Output switches are activated by the ShopBot Software, either from the keyboard or from within a Part File, using the [SO] Command. You would use: [SO] switch number, 1 for ON or 0 for OFF. Switch number is the number from 1 to 4 of the switch you want to turn on or off. You can also hold down the Alt key and use the number keys on the top of the keyboard to toggle an output between OFF and ON. Output Switch #4 is a special case. By default, it is automatically activated when your tool is active, coming ON at the time the Warning Signal (see [SW]) begins to flash and beep. It could be used to activate a relay that controls other processes, such as external warning signals or accessories such as dust collectors. This function for Switch #4 can be adjusted or disabled (see Command Reference section on [SO]).

Additional Axes, Rotary Indexers, Lathes

ShopBots are available with an optional (4th) and (5^{th}) axis. These additional axes represent the electronic hardware for controlling additional stepper motors. The necessary software for the additional channels is made available in the Control Panel. We refer to these extra axes as the A {Accessory) and B axes. They are fully controllable from the software with Commands similar to those of the other three axes (e.g. [MA] to Move the Accessory Axis) or M4 to move the A axis simultaneously with the other 3 axes. Similarly MB would move the B axis and M5 would move five axes simultaneously.

One use of the Accessory Axis is to control a second Z-Axis with router or other tool. The second Z axis can be used to position a different cutting tool up or down (this has the effect of giving your ShopBot a tool changer with two cutters). You can also use a second Z Axis to provide tandem cutting of duplicate parts, with the spacing between parts being based on the physical separation of the two Z Axes.

A second use of the Accessory Axis is as a more general stepper motor controller for use in other types positioning tasks. One example of this use is having the Accessory Axis to drive a rotary axis that is used for indexing a cylindrical piece or a piece for turning work. The B axis can also be used for this purpose. With an appropriate Part File this kind of set up will allow you to do decorative turnings. An Indexer is available from ShopBot as an optional accessory. There are numerous additional potential uses of this additional axis from controlling the angle of the plunge axis, to feeding material, to controlling coolant delivery, etc.

For more information see the *ShopBot Indexer* in the accessories section of the manual.

3-D Digitizing (Probing Surfaces)

One very useful accessory is what is called a probe or digitizer. Your ShopBot can use such a device to sample the 3-D surface of an existing part to create a cutting file. The basic idea is to systematically sample a surface by making a series of Z-Axis plunges. A plunge is made until contact with the surface occurs. The location data is recorded for the contact point and the probe is pulled up, moved to the next location, and the next plunge is made. The contact is sensed by a switch closure or electrical contact. The probe can be as simple as a stylus attached to a micro switch, or, in the case of a metal part, just the electrical contact of a stylus with the part if the part is electrically connected to an Input and the stylus grounded to the tool. A full description of the ShopBot 3-D digitizing probe is available in the accessories section of the manual.

 Copyright 2008 ShopBot Tools, Inc

The easiest way to use the probe is with the ShopBot Virtual Tool called “Copy Machine” accessed with the [**TC**] Command. The Copy Machine helps set up a probing session for either 3D probing or a 2D perimeter digitization. It allows saving the results to an .sbp Part File for immediate cutting, or as a .dxf file that can be brought into a design program for further processing.

Working with G-code

Though we focus on using the OpenSBP (.sbp) files with ShopBots, your ShopBot software will also run standard G-code natively. While OpenSBP files have many advantages in terms of ease of use and customizability, there are times when it can be useful to be able to run legacy code (i.e. from older CAM programs). To do so, just follow the same steps you would use to run an .sbp file. Type [**FP**]; then in the file selection window change the file extension type to the G-code extensions or all. Next select your file and hit Start. The file will run and preview just as if it were an .sbp file. Note that you may even intermix OpenSBP code and G-code in the same file. This means you can easily edit a G-code file using the friendlier .sbp syntax, or cut-and-paste a G-code function that you like into an OpenSBP file.

Running G-code Straight or Converting It. The ShopBot software actually provides you two options for working with G-code files. You can run them directly as described above, or you can convert the file [**FP**] to create a permanent version in the .sbp format. Conversion only takes a few seconds and provides you with a version of the file that is easier to read and manipulate should you need to make any changes to it. If you are in the situation where you have a number of old G-code files that you frequently use you may find it helpful to convert and work with them in the OpenSBP format and not have to worry with the awkward conventions of the older format. On the other hand, if you are frequently generating G-code files, say from and older CAM or design program, then just running the G-code directly is probably the most efficient option. Note that most current CAD/CAM software will output to OpenSBP format files which allows you to work directly with our more straightforward file format.

Using G-code at the Keyboard. For those who have previous CNC experience with G-code formats and syntax and prefer to communicate with a CNC tool in G-code, The ShopBot [**SI; S**ettings Line **I**nput] allows direct control of the tools with G-code. With [**SI**] you can enter one or more G-code commands, edit them if desired, and then run them by hitting Start. The code can also be saved for re-use. This allows you to work live with G-code. The same running/editing/saving features of [**SI**] can also be used with ShopBot Commands.

G-code Syntax and Formatting Considerations. One of the difficulties of G-code is that it comes in many variations and many tools use their own unique version. This means that design software has many different “Postprocessors” for G-code formatted output files and it is sometime difficult to select the correct one. With the ShopBot Control Software, most G-code instructions that are appropriate for use with a ShopBot have been implemented. ShopBot use of G-code is modeled to the NIST CNC standards (NISTIR 6556; published 8/17/2000). Generally speaking, when you choose a G-code output (postprocessor) format in your design software, select a standard FANUC format and you should be fine (if given options in the setup, use line numbers). Note that some G-code instructions, such as G17/18/19 (plane assignment) and G41/42 (tool compensation) do not have ShopBot equivalent commands and are not converted. Many complex functions once generated in G-code routines are now handled better by the CAM program itself or are more flexibly managed in OpenSBP files. There are some differences in how certain functions are handled between OpenSBP and G-code and these are noted below.

Copyright 2008 ShopBot Tools, Inc

Speeds. In the G-code files feed rates are expressed in inches per minute (IPM). OpenSBP uses either inches per second (IPS) or mm per second (mmPS). The translations is handled automatically, but you should be aware of the different standard. Modern tools move considerably faster than when the G-code system was developed and it is much easier to think about motion in terms of displacement over a second rather than over a minute.

Redundant Expressions. There are a few expressions that have different meanings in OpenSBP vs G-code (e.g. M3, Z2). If you use the option of either using line numbers with G-code, or beginning all lines with a G-code, all redundancies are reliably dealt with since the G-code entries are obvious. If you have a file without line numbers that may contain redundancies, you should use the convert function [**FC**] to process the file. The conversion process handles all code as G-code and will resolve any redundancies. Alternatively you can comment out (with an apostrophe, " ' ") any questionable line, or simply put a dummy line number (e.g. N10) in front of the entry, then run the file directly in the ShopBot software.

Supported G-code commands are as follows:

G00 Rapid Move
G01 Cut move
G02 Arc CW (I,J relative always)
G03 Arc CCW (I,J relative always)
G04 Dwell
G28 Goto to Home
G29 Return from home to current position (before G28 command)
G90 Absolute coordinates
G91 Relative coordinates
G92 Reposition/reset Origin

Supported M-code commands are as follows:

M00 Program Stop (implemented as a Pause)
M01 Optional Program Stop (implemented as a Pause)
M02 Program End
M03 Spindle On (Switch 1 ON)
M05 Spindle Off (switch 1 OFF)
M08 Coolant On (switch 3 ON)
M09 Coolant Off (switch 3 OFF)
M10 Clamp on (Switch 5 ON)
M11 Clamp off (switch 5 OFF)
M30 Program End

G-code and M-Code Command Parsing Conventions:

Spacing between coordinates labels and parameters does not affect commands.

For most dependable interpretation, an "N", "G", "X", "Y", "F" or ">" label should appear as the first character on each line. Formatting will otherwise not affect processing of instructions.

Example:

N1G00X1Y1	...no spacing OK
N2 G01 X1 Y2 Z-.25	...spacing between axis labels is OK
N3 G01 X 2 Y 3 Z -.5	...spacing between parameter and axis label is OK

Copyright 2008 ShopBot Tools, Inc

Block numbering is recommended but not required. Mixing numbers with non- numbered blocks is allowed.

Example:

```
G90                 ...set absolute mode (no block number)
N1 G00 X1 Y1        ...block number included on line
G01 X1 Y2 Z-.25     ...no block number
X2 Y2               ...no block number with modal command supported also
N2 G00 X0 Y0 Z1     ...block number/no block number mixing ok
```

Modal G-code functionality is supported.

Example:

```
N1 G90              ...set absolute mode
N2 G00 X1 Y1        ... rapid move (jog) the machine to coordinate X1,Y1
N3 X2               ...rapid move (jog) to X2,Y1
N4 Y0               ...rapid move (jog) the machine to X2,Y0
```

G-code files are limited to on G instruction and one M instruction per line. In the event that there are two instructions, the first G-code on the line will be the one processed.

Example:

```
N1 G80 G00 G90 X1 Y1   ...only do the G80 instruction (G00 and G90 are ignored)
N2 G00 G91 X2 Y2       ...only do the G00 instruction (G91 is ignored)
```

A ‘pass through’ character is also implemented for use is converted files. If you need to issue a ShopBot specific command, begin the line with a ">" (greater than sign). This will pass the line through the converter, less the ">" sign. When directly running a file, OpenSBP-code and G-code can be intermixed.

Example:

```
N1 G90
>SC, 1            ... Pass "SC,1" through unchanged (set smooth cut)
N2 G00 X1 Y1
N3 G00 Z1
```

Parameter order does not matter, but G-code and/or M-codes SHOULD always be first to avoid confusion.
NOTE: If the line does not begin with an N, G, M, X, Y, Z, F, or >, the line will be commented out!

Example:

```
N1 G00 X1 Y1 Z1 F50       ...standard order of commands
G00 Z0 N2 Y0 F50 X0       ...will work but should be avoided, as it is bad form
```

Programming comments are preserved. The default comment header is an open parentheses "(". A percentage sign (%) will also be commented out (as this is the standard G-code file header).

Example:

```
%                              ...standard G-code header will be commented out
(PROGRAM NAME – TestG.TAP  ...program info  commented out
(DATE, Day-Month-Year - 09-01-98 TIME, Hr:Min - 09:14 )
(Use a ½" router bit )         ...useful information for later
```


Copyright 2008 ShopBot Tools, Inc

M-codes are supported on their own line and after G-code commands, but only ONE M-code is allowed per line. M-code instructions will be processed AFTER G-code instructions. If the M-code instruction must be issued first, put it on a separate line BEFORE the G-code instruction. When running G-code directly in the ShopBot software, some M-codes may be redundant if not preceded by a line number.

Example:
N1 M8
N2 G00 X1 Y1 M03 ...M-code processed AFTER G-code instruction
N3 M9 G00 X0 Y0 ...M-code still processed AFTER G-code instruction
N4 M4 M30 ...only M4 will be processed (M30 will be ignored)

Programming Feed Rate. The feed rate is specified in G-code in INCHES PER MINUTE. The program will convert this to ShopBot's format of inches per second (IPS). The feed rate is processed BEFORE other commands (G-code or M-code). If it appears on a line containing a rapid (Jog) command, then the rapid feed rate is modified, but the cut (Move) rate is not changed. If a feed rate appears on the same line as a cut command (CUT/ARC/DRILL) then the cut (Move) feed rate is changed, but the rapid (Jog) rate remains the same.

NOTE: Feed rate command issued after a RAPID command only sets the feed rate of RAPID commands. A feed rate command after any other command will ONLY affect CUT/ARC/DRILL feed rates.

Example:
N1 G90
N2 G00 X1 Y1 F50 ...set JOG to 50 inches/min (.8333 IPS) then rapid (Jog)
N3 G01 X0 Y0 F30 ...set MOVE to 30 inches/min (.5 IPS) then cut (Move)
N4 G02 X0 Y1 I0 J-1 F50 ...set MOVE to 50 inches/min (.8333 IPS) then cut arc
N5 F60 ...set MOVE to 60 inches/min (1 IPS)

G-code Format Reference
G00 X_ Y_ Z_ F_* [MODAL]
The G00 instruction is used to rapidly move the tool/router from position A to B. This instruction should NEVER be used to cut material. This instruction is modal. Once G00 is issued, subsequent instructions only require coordinates (see example). Feed rate issued after G00 instruction only affects G00 instruction. G00 is also affected by G90 absolute and G91 relative mode instructions.

NOTE: *The use of FEED RATE on G00 is nonstandard. ShopBot allows changing of rapid feed rates and this allows that feature to be supported.

Example:
N1 G90 ... set absolute coordinate mode
N2 G00 X1 Y1 F50 ... rapidly move to absolute 1,1 at a feed rate of .8333 IPS
N3 X2 Y2 ... rapidly move to absolute 2,2 at .8333 IPS
N4 G91 ... change to relative coordinate mode (G00 no longer modal)
N5 G00 Y1 Z1 ... rapidly move to 2,3,1 at .8333 IPS

G01 X_ Y_ Z_ F_ [MODAL]
The G01 instruction is used to move the tool/router in three dimensions. This instruction is used to cut material. This instruction is modal. Once G01 is issued, subsequent instructions only require coordinates (see example). Feed rate issued after G01 instruction effects ALL cutting instruction (including ARC, DRILL, and PECK DRILL). G01 is effected by G90 absolute and G91 relative mode instructions.

Copyright 2008 ShopBot Tools, Inc

Example:

```
N1 G90                ... set absolute coordinate mode
N2 G00 X1 Y1 F50      ... rapidly move to absolute 1,1 at a feed rate of .8333 IPS
N3 G01 Z-.25 F30      ... move cutter into material -.25 inches deep at .5 IPS
N4 X2 Y2              ... move cutter diagonal to 2,2,-.25
N5 G00 Z0             ... rapidly move cutter to 2,2,0 at .8333 IPS (retract cutter)
```

G02 X_ Y_ I_ J_ F_ [MODAL]

The G02 instruction is used to move the tool/router in a clockwise motion. This instruction is used to cut material. This instruction is modal. Once G02 is issued, subsequent instructions only require coordinates (X or Y and I and J). Feed rate issued after G02 instruction affects ALL cutting instruction (including MOVE, DRILL, and PECK DRILL). G02 is affected by G90 absolute and G91 relative mode instructions ONLY in the X and Y axis. I and J coordinate are ALWAYS relative to X and Y.

NOTE: The examples below make reference to 0, 90, 180 and 270 degrees. 0 degrees is 3 o'clock on a watch face, 90 degrees is 12 o'clock, 180 is 9 o'clock and 270 degrees is 6 o'clock.

Example:

```
N1 G90                ... set absolute coordinate mode
N2 G00 X2 Y2 F50      ... rapidly move to absolute 1,1 at a feed rate of .8333 IPS
N3 G01 Z-.25 F50      ... move cutter in Z axis to -.25 at feed rate of .8333 IPS
N4 G02 X3 Y1 I0 J-1 F30 ... move cutter in an arc clockwise 90 degrees at .5 IPS
N5 G00 Z1             ... rapidly retract the tool
N5 G00 X2. Y0.        ... rapid the tool to 2,0
N6 G01 Z-.25          ... move cutter in Z axis to -.25
N7 G02 X1. Y1. I0. J1.  ... move cutter in arc from 270 degrees, ending at 180 degrees
```

G03 X_ Y_ I_ J_ F_ [MODAL]

The G03 instruction is used to move the tool/router in a counterclockwise motion. This instruction is used to cut material. This instruction is modal. Once G03 is issued, subsequent instructions only require coordinates (X or Y and I and J). Feed rate issued after G03 instruction effects ALL cutting instruction (including MOVE, DRILL, PECK DRILL). G03 is affected by G90 absolute and G91 relative mode instructions ONLY in the X and Y axis. I and J coordinate are ALWAYS relative to X and Y.

NOTE: The examples below make reference to 0,90,180 and 270 degrees. 0 degrees is 3 o'clock on a watch face, 90 degrees is 12 o'clock, 180 is 9 o'clock and 270 degrees is 6 o'clock.

Example:

```
N1 G90                ... set absolute coordinate mode
N2 G00 X3. Y1. Z2.    ... rapid move to 3,1,2
N3 G01 Z0. F50        ... move in the Z axis to 0 at a feed rate of .8333 IPS
N4 G03 X2. Y2. I-1. J0. ... move cutter in an CCW arc from 0 degrees to 90 degrees
N5 G00 Z2.            ... rapidly move cutter in the Z axis to 2
N6 G00 X1. Y1.        ... rapidly move cutter to 1,1
N7 G01 Z0.            ... move in the Z axis to 0
N8 G03 X2. Y0. I1. J0.  ... move cutter in a CCW arc from 180 degrees to 270 degrees
N9 G00 Z2.            ... rapidly move cutter in the Z axis to 2
```

G04 P_ [NON-MODAL]

The G04 instruction is used to make tool dwell in one location for P seconds. This instruction is sometimes used to give a better finish at the beginning or end of a cut (such as drilling).

Copyright 2008 ShopBot Tools, Inc

It is also used for tools that require start up time/shut down time (i.e. a spindle/torch/plasma cutter that is controlled by the ShopBot). G04 is non-modal and must be entered each time a pause is required.

Example:

```
N1 G90              ... set absolute coordinate mode
N2 M03              ...  turn on switch 1
N3 G04 P2           ... wait 2 seconds for tool to come up to speed.
N4 G00 X1 Y1 Z2     ...  rapidly move to 1,1,2
```

G28 [NON-MODAL]

The G28 instruction is used to make the tool go to the "HOME" position. The home position is 0,0 ABSOLUTE. G90/G91 do NOT affect this command. When this instruction is issued, it saves the current location in three variables in ShopBot format. This allows for G29 return to the exact same position it left when the G28 instruction was issued. This instruction is used to return the machine home when a program is finished, for changing a tool, or for moving the Z axis mechanism out of the way while material is being changed or cleanup is being performed.

WARNING: If you issue G28 back to back, the return coordinates will be lost. Upon a G29 instruction, the program will return to the last spot the G28 instruction was issued, which being HOME (0,0) the machine will start up from 0,0; not the true return coordinates.

NOTE: No parameters are accepted with this instruction. Only HOME (0,0) is used.

Example:

```
N1 G90      ... set absolute coordinate mode
N2 M05      ... turn switch 2 off
N3 G28      ... go to HOME location
N4 M01      ...  program stop (wait for key to be pressed)
```

G29 [NON-MODAL]

The G29 instruction is used to make the tool return to from the "HOME" position. The home position is 0,0 ABSOLUTE. G90/G91 do NOT affect this command. When this instruction is issued, it returns to the previous position set by the G28 instruction. This allows the tool to return to the exact same position it left when the G28 instruction was issued.

WARNING: If you issue G29 without first issuing a G28 instruction, the tool will most likely go to 0,0; since the return coordinates are not set.

NOTE: No parameters are accepted with this instruction. Only the return coordinates are used.

Example:

```
N1 G90      ... set absolute coordinate mode
N2 M05      ... turn switch 1 off
N3 G28      ... go to HOME location
N4 M01      ... program stop (wait for key to be pressed)
N5 G29      ... return to previous coordinates
N6 M03      ... turn on switch 1
```

G90 [MODAL]

The G90 instruction is used to set the coordinate system to absolute coordinates. This instruction affect G00, G01, G02, G03, G81, and G83.

Copyright 2008 ShopBot Tools, Inc

Example:
N0 G28 ... go to HOME (0,0)
N1 G90 ... set absolute coordinate mode
N2 G00 X1 Y1 ... rapidly move to coordinates 1,1
N3 G00 X2 Y2 Z1 ... rapidly move to coordinates 2,2,1
N4 G00 X1 Y1 Z2 ... rapidly move to coordinates 1,1,2

G91 [MODAL]
The G91 instruction is used to set the coordinate system to relative coordinates. This instruction affects G00, G01, G02, G03, G81, and G83.

Example:
N0 G28 ... go to HOME (0,0)
N1 G91 ... set relative coordinate mode
N2 G00 X1 Y1 ... rapidly move to coordinates 1,1
N3 G00 X2 Y2 Z1 ... rapidly move to coordinates 3,3,1
4 G00 X-1 Y-1 Z2 ... rapidly move to coordinates 2,2,3

G92 X_ Y_ Z_ F_ [MODAL*]
The G92 instruction is used to relocate the origin of each axis. G92 does not require coordinates (see example), but issuing it without coordinates causes both the X axis and then Y axis to be assigned 0 (zero). The coordinates are always ABSOLUTE values. Once an axis has been assigned a value, there is no instruction to restore the axis to its previous value. Use this instruction with caution. Feed rate issued after G92 instruction affects ALL cutting instruction (including ARC, DRILL, and PECK DRILL).

WARNING: Be VERY careful with this instruction. This can easily cause the machine to exceed its cutting area!

NOTE: *While traditionally not a modal instruction, in this implementation it is modal.
NOTE: The best way to return the reassigned origin to the HOME position is to issue a G28, issue a G92 then issue a G28 again. The second G28 resets the return coordinates to 0,0. See G28 instruction for more information.

Example:
N1 G90 ... set absolute coordinate mode
N2 G00 X10 Y10 F50 ... rapidly move to absolute 10,10 at a feed rate of .8333 IPS
N3 G92 ... set X axis coordinate and Y axis coordinate to 0,0 (not Z)
N4 G92 X1 Y1 Z1 ... set X coordinate to 1, Y coordinate to 1, and Z coordinate to 1
N5 G28 ... rapid return to HOME
N6 G92 ... set X axis coordinate and Y axis coordinate to 0,0 (not Z)
N7 G28 ... reset return axis to 0,0 (precaution if a G29 instruction is issued)

M-Code Format Reference

M00 [Program Stop] Pause
The M00 instruction is used to pause the machine. This could be to clear chips/sawdust from the material or to check the machine/cutter/tool. Resumption of the program will continue once a key is pressed on the keyboard.

WARNING: Avoid accidental contact with keyboard, as machine can start up abruptly without warning.

Example:
N1 G90 ... set absolute coordinate mode

Copyright 2008 ShopBot Tools, Inc

```
N2 G00 X1 Y1 F50    ... rapidly move to absolute 1,1 at a feed rate of .8333 IPS
N3 X2 Y2            ...  rapidly move to absolute 2,2 at .8333 IPS
N4 G00 Z2           ... rapidly move Z axis to 2
N5 M00              ... stop program until a key is pressed
```

M01 [Optional Program Stop] Pause
The M01 implemented as M00 (see above). Both M01 and M00 can be used interchangeably.

WARNING: Avoid accidental contact with keyboard, as machine can start up abruptly without warning.

Example:

```
N1 G90              ... set absolute coordinate mode
N2 G00 X1 Y1 F50    ... rapidly move to absolute 1,1 at a feed rate of .8333 IPS
N3 X2 Y2            ... rapidly move to absolute 2,2 at .8333 IPS
N4 G00 Z2           ... rapidly move Z axis to 2
N5 M01              ... stop program until a key is pressed
```

M02 [Program End] End
The M02 instruction is used to end a program. This instruction is generally used at the end of a program to indicate that the machine will receive no more instructions.

WARNING: Avoid contact with keyboard, as accidental resumption of program can occur.

Example:

```
N1 G90              ... set absolute coordinate mode
N2 G00 X1 Y1 F50    ...  rapidly move to absolute 1,1 at a feed rate of .8333 IPS
N3 X2 Y2            ...  rapidly move to absolute 2,2 at .8333 IPS
N4 G00 Z2           ... rapidly move Z axis to 2
N5 M02              ... end of program
```

M03 [Spindle On CW] Switch #1 ON
The M03 instruction is used to turn SWITCH #1 ON.

M05 [Spindle Off] Switch #1 OFF
The M05 instruction is used to turn SWITCH #1 OFF.

Example:

```
N1 G90              ...  set absolute coordinate mode
N2 M03              ...  turn on SWITCH #1
N3 G04 P2           ...  pause for 2 seconds (for machine to start up)
N4 G00 X10 Y10 Z.25 ... rapidly move to X 10, Y 10, Z .25
N5 M05              ...  turn off SWITCH #1
N6 M02              ... end of program
```

M08 [Coolant On] Switch #3 ON
The M08 instruction is used to turn SWITCH #3 ON. This can be used to control anything such as dust collector or spray mist for cooling cutter/work.

M09 [Coolant Off] Switch #3 OFF
The M09 instruction is used to turn SWITCH #3 OFF. This can be used to turn off a dust collector or spray misting unit.

Copyright 2008 ShopBot Tools, Inc

WARNING: Use these instructions carefully, as accidental startup of devices can occur without warning. Use of relays to control tools is STRONGLY discouraged.

Example:

```
N1 G90                  ... set absolute coordinate mode
N2 M03                  ... turn on SWITCH #1 (turn on tool)
N3 M08                  ... turn on SWITCH #3 (turn on dust collector)
N4 G04 P2               ... pause for 2 seconds (for machine to start up)
N5 G00 X10 Y10 Z.25     ... rapidly move to X 10, Y 10, Z .25
                  ----- perform machining operations -----
N97 M05                 ... turn off SWITCH #1 (turn off tool)
N98 M09                 ... turn off SWITCH #3 (turn off dust collector)
N99 M02                 ... end of program
```

M10 [Clamp On] Switch #5 ON

The M10 instruction is used to turn SWITCH #5 ON. This can be used to control such things as vacuum hold downs etc...

M11 [Clamp Off] Switch #5 OFF

The M11 instruction is used to turn SWITCH #4 OFF. This can be used to turn off such things as vacuum hold downs etc...

Example:

```
N1 G90                  ...  set absolute coordinate mode
N2 M03                  ... turn on SWITCH #1 (turn on tool)
N3 M010                 ... turn on SWITCH #5 (turn on vacuum hold downs)
N4 G04 P2               ... pause for 2 seconds (for machine to start up)
N5 G00 X10 Y10 Z.25     ...  rapidly move to X 10, Y 10, Z .25
                  ----- perform machining operations -----
N97 M05                 ... turn off SWITCH #1 (turn off tool)
N98 M11                 ... turn off SWITCH #5 (turn off vacuum hold downs)
N99 M02                 ... end of program
```

M30 [Program End] End

The M30 instruction is used to end a program. This instruction is generally used at the end of a program to indicate that the machine will not receive any more instructions. Interchangeable with M02. An END instruction is optional.

Example:

```
N1 G90                  ... set absolute coordinate mode
N2 G00 X1 Y1 F50        ... rapidly move to absolute 1,1 at a feed rate of .8333 IPS
N3 X2 Y2                ... rapidly move to absolute 2,2 at .8333 IPS
N4 G00 Z2               ... rapidly move Z axis to 2
N5 M30                  ... end of program
```

Working with the ShopBot Converters for .dxf, .hpg/.plt, .bmp, and .jpg

If your design software does not directly generate files in the ShopBot Part File format, you can quickly convert them to Part File format using one of the built-in converters. The conversions take place quickly and you will hardly notice the additional step in getting to the Part File representation of your project.

 Copyright 2008 ShopBot Tools, Inc

NOTE: Many vector format files (.dxf, .eps, .ai, .pdf) can be imported into ShopBot's **PartWorks** design software where further work can be done on them. Bitmap files (.bmp, .jpg, .gif, .tif) can be imported for automatic or manual tracing. Three dimensional models (.stl, .v3m, .3ds, .dxf, .wrl) can be imported into **PartWorks 3D** for generation of Part Files. Bringing files into one of the PartWorks design programs has become the preferred method of converting files from other software because it provides full control of the tool-pathing process as well as visualization of the finished part or project. The following conversion techniques are most useful when you need a quick conversion, or have a process for file generation in other software and are comfortable with producing a direct toolpath from it.

Starting a Conversion

You use the [**FC**] Command to start the conversion of a file. Select the file from the display and hit ENTER. If you want you can filter by file type by clicking on the arrow by the file type box. If the file type is recognized the appropriate converter will be started. You may then be asked a few additional questions -- usually questions that help define the plunge depths for the Z Axis. After all the questions have been answered, the converter will run and a ShopBot Part File will be generated. The conversion usually takes only a few seconds. The new Part File will have the same name as the file you converted, but with a '.SBP' file extension except when you are converting a .sbp file to a .dxf.

The generated Part File will be a standard file OpenSBP Part File. You can open it with the editor [**FE**] and inspect, document, or modify it. And of course, it's ready to run in Preview Mode or Cut Mode. We do suggest that you always preview or 'air cut' a new Part File to make sure it is functioning properly before cutting material.

TIP: You will usually find it helpful to put a little documentation in the top of your converted Part File that indicates what the file is for and any specific information that may be needed for setting up the part for cutting. See the Working with Part Files section above for details.

Each converter can be run as a standalone processor if desired. The path to each of the converters is C:\Program Files\ShopBot\ShopBot 3\Bin.

Specific Info for Each Type of Conversion

The details for each type of conversion are a little different because the initial file types contain different types of information. In the following sections we provide details on the specific issues related to each type of conversion process.

Conversion from .DXF Format:

Most typically, you will find yourself using the conversion from .DXF format because this is the standard 'vector' drawing format. The universal nature of the .DXF file format makes it possible for ShopBot to work with virtually every type of CAD and vector drawing format (2D or 3D). In a 2D drawing, you will be able to use different line colors to define the plunge depth for the Z Axis.

There are a couple limitations to using DXF files from a CAD program that we have already noted in the CAD section above. Briefly, the .DXF format is not intrinsically oriented to cutting or machining a part. While it will faithfully translate the exact location of all lines, your CAD software is unlikely to worry about putting the cuts in the order that you want to cut them when it generates the .DXF file. Typically, the lines are saved in the order that you draw them. Thus you will need to plan the drawing (or do a retracing of it) so that the segments are entered in the order that you want to cut them.

Copyright 2008 ShopBot Tools, Inc

A second limitation of .DXF is that it knows nothing about your cutter and does not take the diameter of the bit into consideration in generating the file. You will need to plan the 'cutter compensation' into the drawing that you do – the drawing represents the center of the path of the cutter.

You will also find that some CAD objects or entities need to be 'exploded' or 'vectorized' before they can be converted to a Part File. True-Type fonts are one example. These fonts need to be exploded to their line segment components so that they can be used to generate a tool path. Curves (splines) are another type of entity that will need to be exploded. Each CAD system has a different technique for doing these explosions, but in most cases it is just a matter of selecting the entity and clicking on one function.

Because of the limitations, in most cases it is preferable to bring the .DXF file into ShopBots PartWorks design software where CAM functions can be used to get the tool paths just the way you want them.

But for some situations and with these above limitations in mind, the .DXF conversion process can be productive and straightforward method to get cutting. You can deal with its limitations by carefully planning the drawing and by keeping in mind that you are creating the cutter path ... not just the shape of the part. Typically, you will want to offset the cutting path to the outside of the part you want cut by one half the diameter of the cutter. For a simple drawing, this is not too difficult, and can usually be accomplished with the function for creating a parallel line that is available in most CAD programs. For example, you would draw the part you wanted to cut out, then using the parallel function you would draw a line around it that is parallel and spaced one half the diameter of the cutter. It is this second line that would be saved to the .DXF file and loaded into ShopBot. Some CAD programs offer macros or some form of customizability that will make such cutter paths easier to draw.

Making Use of .DXF. Use your CAD or vector drawing program (such as Corel Draw) to draw the tool path you want your ShopBot to follow. If you want to have different Z levels of plunge/cut, enter these cuts in a different color in your CAD program. The converter will prompt you for a depth for each color during the conversion process. The converter directly deals with lines, polylines, circles and arcs. Curves or splines need to be changed to a vector/segmented form in the CAD program before they are saved. Different programs call this segmenting process by different names ... typically something like 'explode' or 'convert to vector.' True-Type Fonts also need to be saved in their segmented form. After segmenting, save this drawing as a '.DXF' file (in some CAD programs you will use 'Export' rather than save). Then start the ShopBot software and use the [FC] Command to convert the .DXF file to a Part File. It is usually most convenient if you save the file into your Part File Directory -- this way, the drawing and the converted file will be kept together.

During the Conversion. At conversion time, you will be asked if the file is 2D or 3D because the software wants to know how to handle the Z axis. For a 3D file, the Z axis motion is obtained from the .DXF file. If you designate 2D, then the Z Axis plunges are determined by line color. So for a 2D file, your ShopBot needs to know how deep the cut is to be. When the .DXF file is being read in, you will be prompted for a depth for every color of line (or layer) that you used in your drawing. These depths are translated into Z axis motions and put into the ShopBot .SBP file as it is being generated.

If you are doing frequent DXF conversions from the same drawing program, you can speed up the conversion process by creating a set of predefined responses to the color-depth question. You do this by editing the UX_DXF.INI file. The path to this file is C:\Program

 Copyright 2008 ShopBot Tools, Inc

Files\ShopBot\ShopBot 3\Bin\ UX_DXF.INI. You can change the values to the existing colors or you can add additional colors and depths.

TIP: Each CAD program uses different numbers for colors. You can figure out the color numbering for your CAD program by drawing some entities in different colors, then running the converter to see what color numbers are found and which is associated with which entity. The basic color numbering for AutoCAD is:

0-Black	5-Blue
1-Red	6-Magenta
2-Yellow	7-White
3-Green	8-Grey
4-Cyan	9-Light Grey

The numbering for Corel Draw (default Version 8) is:

0-Black	22-Blue
15-Red	16-Magenta
14-Yellow	10-White
13-Green	12-Cyan

Conversion from HPGL Plotter File Format:

The HPGL converter is designed to convert HPGL 7075A plotter commands to ShopBot Commands. HPGL is a machine control language used mainly to control plotters, vinyl cutters, and some engraving machines. HPGL stands for Hewitt Packard Graphics Language. The HPGL is a 2 dimensional vector data language with Pen Up (PU) and Pen Down (PD) commands. Pen commands normally control when the pen goes down to draw a line or up to skip over sections. Later, Hewitt Packard implemented a pen carousel. This allowed 'plots' to be in color. Colors are implemented with a Set Pen (SP) plus the color number in the pen carousel. As with the .DXF Converter, different colors can be used to define different depths for cutting with your ShopBot. The conversion that is done here is pretty generic and should work with most HPGL files. But if your software allows you to choose the plotter format that you are exporting to, then select the HP 7075A.

General. To convert HPGL language files to ShopBot format, we have to specify a depth that represents a pen location or color in the carousel. This program implements 11 predefined depths. All depths will be specified in absolute coordinates. When no Set Pen (SP) command is included within the HPGL file, the program automatically defaults to "Color 0". So "Color 0" will be used most.

The HPGL language allows two modes of movement: absolute and relative coordinates. This converter will deal with either type movement, and will appropriately set ShopBot to relative or absolute mode.

NOTE: even though the tool may be in relative movement mode, the depths will ALWAYS be absolute depths. The Pen Relative (PR) command moves the XY carriage relative to its present position. The Pen Absolute (PA) mode moves the XY carriage to absolute coordinates.

During the Conversion. The converter stores settings, so you do not have to re-enter them every time. If there is not a setting file (HPGL2SBP.INI), the program will create one, but ALL depths will be 0. Once you have defined a list of settings, you will not need to edit them again, unless you want to make a change to them. The settings can be edited in the program or with a text editor (such as ShopBot's [**FE**] Command). There is a sample .INI file at the end of this section.

 Copyright 2008 ShopBot Tools, Inc

The program firsts asked for the name of the HPGL file. There is no standardized extension for these files. Programs may use .PLT, .HPG, .PRN, .TXT, or some other extension. The .PLT and .HPG extensions are recognized by ShopBot and the HPGL converter will automatically be started if you choose a file with these extensions. If you use another extension, you will need to answer the 'type' prompt to indicate that you want to attempt an HPGL conversion.

Next, the program will display all the current settings. To change a value, press the letter in front of the label. When you are done making changes or you do not wish to change any of the values, press "Q", [SPACEBAR], or [ESC] key. If any values have changed, the program will ask if you would like to save the new values. Press "Y" to save the new settings or any other key to start the conversion. You will see:

Enter name of HPGL file to convert (+ extension):..\test.plt
==========

- C] ConversionFactor = 1016
- D] DefaultScale = 1
- S] SafeHeight = .1
- 0] Color0 =-.125
- 1] Color1 =-.01
- 2] Color2 =-.02
- 3] Color3 =-.03
- 4] Color4 =-.04
- 5] Color5 =-.05
- 6] Color6 =-.06
- 7] Color7 =-.07
- 8] Color8 =-.08
- 9] Color9 =-.09
- A] Color10 =-.1

==========
Press the key that correspond to the parameter you want to change or Q to Exit
==========
Starting Conversion...
Finished...

In the HPGL coordinate system, there are 1016 "units" to an inch (or 40 "units" to a millimeter). This allows a resolution of about .001 inch. Some programs convert the "units" system to a different measurement. The converter allows you to adjust the "units" setting in the conversion if your drawing program uses a different setting. You should not need to change the ConversionFactor setting unless your images do not come out the size that you expected.

The program also allows you to adjust the scale size. For example, (1) will output it at 100%; (2) will create output twice the size (200%).

The conversion makes use of ShopBot's variables. Variables are used to make it easy for you to later change the depths in a converted file. You will see &Zsafe, this is the Safe Height that the bit will retract too. Then each &Z# represents each color. You can make a change here and the entire file will now use the new depth when it is run again. This is also useful when you accidentally specified the wrong depth to the wrong color. The converter also produces a label in the generated file at each point where the depth changes. This is displayed while the file is being cut. This way you will know what depth setting is being used while that part of the program is running.

Copyright 2008 ShopBot Tools, Inc

The HPGL2SBP.INI file holds the default depth settings and consists of variable names in square brackets. The values must be on the right hand side of the equals sign. Comments may be included in the .INI file, but they must have an apostrophe (') at the beginning of each comment line.

Note: If you change the values within the program and save the settings, all comments will be deleted. The label should be self descriptive, but Conversion Factor and DefaultScale are discussed in the Overview Section. SafeHeight is the Safe-Z height above the material that the bit will retract to when the program issues a "Pen Up" (PU) command. This height should be above ALL clamps, holders, and surface of the material itself. Each Color# represents a depth setting, that the converter will use when. Color 0 is the default location, as many programs do not specify a color ["Set Pen" (SP) command] so Color 0 is the default depth for HPGL files that do not include color information. Here's an example of the HPG2SBP.INI file:

- 'Default Configuration File for HPGL2Sbp
- 'Color 0 is default depth, if NO color is specified
- 'All depths are absolute (should be negative numbers)
- 'SafeHeight must always be 0 or greater!

    - [ConversionFactor]= 1016
    - [DefaultScale]= 1
    - [SafeHeight]= .1
    - [Color0]=-.125
    - [Color1]=-.01
    - [Color2]=-.02
    - [Color3]=-.03
    - [Color4]=-.04
    - [Color5]=-.05
    - [Color6]=-.06
    - [Color7]=-.07
    - [Color8]=-.08
    - [Color9]=-.09
    - [Color10]=-.1

We are grateful to ShopBotter Bruce Clark, author of much of the conversion software for ShopBot.

Copyright 2008 ShopBot Tools, Inc

Maintenance

Other than the types of adjustments you did as you set up your ShopBot to align the axes and adjust the motors, there should be little specific maintenance required for your ShopBot. Here are a few things to periodically check to make sure your ShopBot is working at its best. A couple of general points first:

LUBRICATION: The rack and pinion system works best if greased. We do not find that this is problematic in terms of collecting debris. Use any regular bearing grease and periodically renew it. A light protective film of oil or specialty coating will help protect the rails from rusting in a humid environment.

ShopBots are fairly immune to sawdust, but you do want to keep the tracks and the racks relatively free of dust and debris. An occasional blast of air should help keep things clear. Using a dust collection system will further reduce the likelihood that dust build-up will hamper the action of your tool.

Mechanical Adjustment - X & Y Axes

1. Make sure that all the X & Y wheel bearing are properly seated on the tracks -- make sure that you check at several places on the track. Check the movement of the axis by pushing the carriage ... it should move smoothly. If in doubt, release the pinions from the rack and push the cars around. **NOTE: The motor cables should be disconnected from the drivers before you push the tool with the motors engaged in the rack.**

2. Power up your tool and move it a around a bit. Then try to wiggle each carriage near the its motor. You are testing to make sure there is no looseness in the drive train -- that is, you want to make sure the pinions are not be slipping on the motor shaft and that it is fully engaged into the rack. You may need to tighten the set screws in the pinion, or tighten the motor bracket to more engage the pinion.

TOLERANCE BUILD-UP

Numerous small problems of tune and adjustment, over time, can contribute to "tolerance build up" and start showing up in the quality of your cutting or machining. Here's a list of the things to keep an eye on:

1. Your step resolution is ~.0005. Keep in mind that this number represents the absolute best that your tool could do for accuracy, and it means that every step it takes is a digital move of this length. This digital stepping can show up on diagonals and curves.

2. Check the collet - it should be tight, clean and in good shape. High mileage collets belong in the trash. Good quality spring collets advertise a life of <700 hours. As the collet goes bad, it no longer holds the bit correctly, and it will cause the router to vibrate from asymmetry.

3. Check you bit. Dull and out of balance bits are the most frequent cause of bad cuts from chatter and uneven movement. Improperly sized bits can also increase deflection. The type of bit you choose can also have a good or bad effect on edge quality. Your tool has a lot of power, but you don't want heavy forces put on the bit because it will flex the bit and your tool and put a serious strain on the cutting process.

Copyright 2008 ShopBot Tools, Inc

4. Check the run-out (side to side) and end play (up and down) in your router bearings with a dial indicator. You will find most new off the shelf routers with a range of .004"-.006" in both directions. Unless you spring for a spindle, you will need to live with this run-out. Be aware that as the bearings wear, this run-out will get worse. When the run-out gets bad, replace worn bearings with new ones.

5. Make sure your part is held down properly. $100,000 CNC's that hold very high tolerances still have a ¼" error when their part slips ¼". Constantly evaluate how you are holding your parts. Consider a special jig or vacuum box. A vacuum table will not only hold the part horizontally, but will prevent it from vibrating up and down.

6. Check the Z axis: Grab the Z-axis at the base of the router (please unplug) and apply pressure along the X and Y-axis separately with the stepper motors powered up.

   *The Y axis car has eccentric adjustment nuts on the two lower bearings and may need occasional adjustment. Rotate each nut slightly, counter-clockwise to tighten the Y car up.

7. Now check for up and down play that can come from backlash on the rack and pinions. The motor bracket can be tightened to better engage the pinion

8. Next, we can check for any play in the X- and Y-axis carriages that might indicate a loose set-screw and pinion that is rotating slightly on the motor shaft. With the motor on and the tool not moving, try to push the carriages back and forth. They should be locked in position. [As Per Above]

Electrical/Electronic

There should be no maintenance of electrical or electronic parts required. Make sure all the connectors stay firmly in place. Do not risk electrical interference with the operation of your tool by adding any more tools or appliances to the ShopBot circuit than necessary.

There are 2 things to avoid to protect your electronics:

1. Never unplug a motor or motor cable with the power on. This can cause a surge of power to the motor driver circuitry which will destroy it (a loose motor wire will do the same thing).

2. Don't push the tool manually/physically with the motors plugged in to the Control Box, whether it is off or on. The motors can generate a current that will zap the motor drivers. (It is ok to push the tool a very short distance, very slowly, as you might do to push it up against the end stops from ½" away.) Just unplug the motor wires before you do any pushing around.

Note: Stepper motors and their electronic drivers in the Control Box will tend to run warm. However, if either should get extremely hot to touch, it is an indication of a malfunction or poor wiring connection.

Note: The motors may hum when they are at stand-still because the power being applied to them is PWM (pulse width modulation) 'chopper' electronics. This sound is normal. There will also be a slight 'thunk' when the motors are initially powered on.

Copyright 2008 ShopBot Tools, Inc

Control Software

ShopBot provides free Control Software upgrades via our website. The current version of the software is always available for download from our Web Site: http://www.ShopBotTools.com (or you can request a CD)

When new software becomes available, you will generally install it over your existing version.

Troubleshooting

The individual components of a ShopBot are pretty simple. That makes them relatively easy to deal with should you have some sort of problem with your tool. Additionally, one of the advantages of having assembled your own tool is that you already are very knowledgeable about its construction and will have a good understanding of where any problem might originate. The difficulty in fixing the occasional problem with your tool arises because even though the components are simple, they work together in relatively complex ways making the diagnosis of a problem sometimes tricky. There is the mechanical (cutter and tool), the electrical, the electronic, and the software which all work together dynamically ... the site of the problem is not always obvious, but figuring it out should be your first concern in troubleshooting.

TIP: The Doctor ShopBot website has diagnostic help for common problems (http://drshopbot.com) and is a good place to get quick help with a problem you may be encountering. There is also a Technical Posts section on the main ShopBot website which contains current Tech Support Bulletins.

To quickly get your tool back in operation, whether you're solving the problem yourself or getting help from ShopBot Technical Support, the important thing is to identify the nature of the problem you are having and figure out at what level in the system it is occurring. It's also important to give some thought to when the problem is occurring and whether it is likely to be a single issue, or multiple little issues that are getting you bogged down. In the following section, we'll give you some structure to help characterize your problem with the idea that this will help focus your thinking on the problem and make it easier for you or us to get you going again.

Copyright 2008 ShopBot Tools, Inc

First get your thinking right …

This is particularly important if you are e-mailing, faxing, or calling ShopBot Technical Support ... we all need to be pointed the right direction to efficiently get to the solution -- and we will. You basically need to figure out whether you are having a file creation or software problem (#1 and #2 below); or are dealing with an operational problem by which the tool not working right, such as when you move it with the arrow keys or give it a simple move command from the keyboard (#3 below). Here's a little help in determining where the problem in …

1. ***File Creation Problem***. Is this a problem with the design software you are using or with getting the design into ShopBot Part File format?

   General suggestions: Make sure you've had a look at the section on 'Getting from your idea to a ShopBot Product' and the section on 'Converters' above. Then communicate the problem to us. It will help if you e-mail a copy of any problem file, both the converted and original version, as well as the PROBLEM.LOG file from your ShopBot directory/folder.

2. ***Software Function Problem.*** Is this a problem with getting the tool to carry out some function (that is, is it the software that does not seem to be working right or the way that you think it should)?

   General suggestion: Make sure you've looked at the details of the Command that you're having problems with in the Command Reference Manual (also available in Help within the ShopBot Software). Check the sample files (those that are S_.... .SBP in your ShopBot directory) for one that may do something like what you're trying to do. If the function is in a file, make certain that you've got the parameters laid out correctly. When none of this helps, communicate a description of the problem to us. E-mailing a copy of the file that does not work, or the way you are entering a Command, is the most efficient way for us to get you feedback on what is happening. Also send the PROBLEM.LOG file from your ShopBot directory/folder so that we can duplicate your configuration.

3. ***Tool Operation Problem.*** Does your tool not run right when you give it an instruction or start executing a file that you believe should work? We would like to be able to know for sure if the problem is an electrical-mechanical problem with tool operation, and rule out software problems. Test the tool with air-cutting or real-cutting using Commands from the keyboard such as circles [CC] or rectangles [CR].(If the problem happens only during the execution of a file, consider that it may be a software issue related to Part File programming or conversion rather than a tool operation problem.)

   General suggestion: In the following sections we'll work to deal fully with Tool Operation Problems. If we don't work it out here, communicate to us what you've figured out so far, along with any relevant files and your PROBLEM.LOG file from your ShopBot folder/directory, and we'll work to get you running quickly... There are two other considerations for Tool Operation Problems.

   1. First, it's important for us to know whether this is a problem that is occurring at your first use of the tool, or whether your tool has been running for a period before the problem occurred.
      -If the latter, what are the conditions under which the problem occurred. Have you changed anything with your tool or in your Shop? Does the tool work for a while and then fail?

Copyright 2008 ShopBot Tools, Inc

2. The second major question regarding Tool Operation Problems is whether this is a primary failure, such as the motors not moving, or a qualitative problem, such as cutting being rough, uneven, or inaccurate. We appreciate that both are important, but in the former case the focus will be on an overall system failure while in the latter it will be on the operation or adjustment of components or assemblies, configurations, and procedures.
   If your problem is a primary failure, let's distinguish a bit further.

   1. **Connection.** Your computer is not getting connected to the ShopBot Control Box or is disconnecting during operation.
   2. **Movement Direction.** One or another axis not moving in right direction.
   3. **Movement/Location.** Axis not going to where it is supposed to go. Does it return to 0, or does it also lose its 0 location?
   4. **Movement/Stalling**. Axis moves but stalls and misses steps.
   5. **No Movement**. One or more axis not running, or nothing moving.

Dealing with Tool Operation Problems: Primary Failures

Problem with 'ShopBot not connecting': If when your ShopBot software starts up it is not able to connect to your ShopBot, let the software lead you through checking for another port. When you instruct it to try, the software will look for your ShopBot on all the available Ports on your computer, one at a time. Once having found your ShopBot, the software will go to the correct port next time you start up.

If your ShopBot connection is not found on any of the available ports, we'd better check a few obvious things first. Is your Control Box plugged into one of the USB or Serial Port connectors on the back of the computer? Check to make sure the ShopBot USB drivers were installed with the software. It may be easiest to just re-install the software and make sure that you check the box to install the USB Drivers that appears in the final dialog window.

Intermittent Connection Problem:

If you receive a 'Connection ...' error or a 'Too many characters received' error, what has happened is that there has been a temporary communication problem between your computer and the Control Box. Typically, your ShopBot will 'Re-connect' to the computer and you can continue operations. A brief power failure in your shop will produce this effect. However, if this problem occurs with any regularity, it is important to figure out what is causing it and deal with that.

This symptom is sometimes the result of screen saver or power management software turning on when a long Part File is running and no keyboard input occurs on your computer. When using a computer to run your ShopBot, it is best to turn all screen savers and power reduction software off. This is usually accomplished in the Windows screen properties settings, but you may need to disable sleep functions in your computers BIOS. If you are using a laptop computer, it must be plugged into power. Otherwise, power will be automatically reduced to the USB port.

Electrical noise generated by other shop equipment such as compressors can sometimes generate voltage surges that interfere with the communication between your ShopBot and the computer. Make sure your tool and control box are both grounded to a system ground (such as a piece of conduit or electrical box). An uninterruptible power supply (UPS) for the computer will usually eliminate such problems. Also, try to separate the power wires for the router and dust-collector as much as possible from the ShopBot wiring, and especially fromthe USB cable between the Control Box and computer.

Problems with Movement

 Copyright 2008 ShopBot Tools, Inc

Changing the direction of a motor

PRS<u>standard</u> ONLY!!! Do Not Alter Connections on a PRS<u>alpha</u>! ALTERING THE CABLES ON A PRS<u>alpha</u> TOOL WILL VOID YOUR WARRANTY!!!!!!!!

It is straightforward to change the direction of a PRS**standard** motor and essentially involves just reversing two wires in the motor cable connector. To change the direction that a motor is running, turn all the power off and unplug the motor cable for the motor from the Control Box (remember that if you are changing the direction of both motors on the X axis, you will need to make the following change to both motor wire connectors, they have one and two red tape bands). We just need to change the position of two wires in the connector (only change the one connector on the motor wire that goes into the Control Box). You should have a little nylon tool that came with your Control Box or documentation. It can be used to open the spring clips in the gray connectors and release the wires then re-insert them (these are Wago Connectors so we call it a WAGO Tool). If you cannot find this tool, you can use a very small screwdriver to release the wires (slide the blade of the screwdriver into the slot just above the hole that the wire goes in; when its all the way back, lever the tip down a bit and the wire will release). Now, to change the direction of a motor, reverse the green and black wire positions in the connector. This would be the wires in pin 3 and 4 on the connector. Plug the connector back in, and off you go ... (if you did both the X motors, better test them one at a time to make sure they are both going the right direction ... and of course, turn the power off each time you plug or unplug a motor).

Determining whether you have a software, electrical, or mechanical problem when an axis moves but does not go the correct distance

Let's try the move a couple of times and over different distances so you're sure of what's going on. Make certain you are in absolute distance mode [SA] and correctly understand how absolute distance is working. That is, if you issue a [MX] 2 Command your tool will go to where it thinks the 2.000 location is. If you issue the Command again, the tool will not move, because it is already at 2.00. You need to give it the Command [MX] 4 to go another 2 inches up the X axis. At this point also check to make sure the speed has not been inadvertently set too high. The XY Move Speed should probably be in the range of .5-4.5 in/sec and the XY Jog Speed at 4-6 in/sec. Try the Z Move Speed at 1.5 and the Jog Speed at 3 in/sec. Stalling (and lost position) will occur if the speed is set too high.

Another thing to check while we're still powered up is to make sure that the set-screws in the pinion gears are tight and the pinions not rotating on the motor shaft. You can test this by trying to push each motor car back and forth with the power on but the tool not moving. The car should be locked in position. If the pinion is loose, you will feel the car slip back and forth.

So, assuming that we've got the idea of the Commands right, the speed is not set too high, and the pinions are not slipping, does the tool seem to be running relatively smoothly and consistently making the same error -- for example, going a little too far or a little short. If so, we probably have a software issue involving setting the correct 'Unit Values' for the gear ratio of your tool. ShopBots use a variety of gear ratios depending on their configuration, thus it is necessary to make sure you've set the Unit Values correctly when you first use the software and when you subsequently upgrade the software. The Command [VU; 'V'alues 'U'nit] allows you to enter the correct parameters for the gear ratio on your tool. You should be able to find those values in the documentation we have sent with your tool. There is one number that will be used for the X and Y axes, and another for the Z. If you have a second Z axis or indexer, you will also need to set the Unit Value for that axis.

Copyright 2008 ShopBot Tools, Inc

A Unit Value difficulty is pretty easy to deal with and does not reflect any kind of problem with the assembly or set-up of your tool. However, if you are experiencing a movement problem in which your tool seems to stall, moves erratically with little power, or moves variable distances and less than the full distance that it has been sent, we need to determine whether you are experiencing mechanical binding or have an electrical problem.

Our approach to isolating the problem will be to first rule out (or in) physical binding. Turn everything off and start by disconnecting all the motor wire(s) at the Control Box. Do not manually move the tool unless the motor wires are disconnected from the control box! Now for a problem with the X or Y axis, pull the carriage gently but firmly up and down the axis. You will remember that the motors create some resistance; however, you should be able to produce a smooth and even movement up and down the track. For the Z axis, you should be able with some force to push the carriage up and down. If you get smooth action on the axis that is malfunctioning, we probably do not have a binding problem.

However, if there is any question of smooth action, we should look a little further. For the X and Y axes, you will need to release the motor pinions from the rack. You should be able to drop them in their slots. With the motor(s) thus removed, re-evaluate the free movement of the axis. If it is binding in a particular location, try to identify the source of the binding. If binding of the carriage does not account for the problem you are having but the carriage was hard to move or moved unevenly before you removed the motor, evaluate whether the motors and pinions may have been pressed too tightly against the rack, may be too loose on the bracket, or may be misaligned. Grease the rack with bearing grease or other heavy lubricant. Lastly, have a look to make sure the motors are aligned square to the rack (if there is a horizontal angle between the pinion and the rack, the pinion will try and 'walk' back and forth in the rack and this will cause binding; if there is a vertical angle, the pinion will bite too deeply and bind). These adjustments should insure the free movement of your tool. If you are still experiencing problems that seem like binding or stalling, there may be a difficulty in getting full electrical power to the motors.

Evaluating the electrical power output to the motors is sometimes a bit tricky in the situation where the axis partially functional. This is because a stepper motor can still partially function even if the electronic driver is damaged or one or two of its power leads are not making a good connection. However, in the case of a bad connection or driver the motor will have low power and will often 'stutter' if a little resistance is applied. So set the Move Speed to 1in/sec and put the problem axis in motion (say with an MX or MY 30). Then grab the car. On a PRSstandard tool there should be 60+ pounds of force at 1in/sec before the cars stalls or slips. A PRSalpha will have about 200 pounds of force at low speeds.

The most likely cause of a power problem is in the connectors or wiring to the motor. The second most likely is a problem with the electrical driver. It is virtually unheard of to have a motor fail.

See if you can isolate the problem by swapping motor wires. Reverse the connections of the two X motors in the Control Box and see whether the problem moves to the other motor. Or plug the Y motor into one of the X motor outputs at the Control Box. Remember, don't unplug or plug in any wires without turning the power off first. Using this strategy, you should be able to determine if the problem is with the driver for the axis in the Control Box or with the specific motor wiring or motor. If it's related to the motor wire or motor, have a close look at the connectors or swap wires with another axis to further isolate. If the problem seems to be with the driver, give us a call and we will help you confirm this test and quickly get a new driver to you if necessary.

Motor not functioning at all: If all axes seem completely dead, it is probably because the high voltage power supply is not functioning right. If a single axis seems dead, with the

 Copyright 2008 ShopBot Tools, Inc

motor not moving or making any noise, and not powered up and locked in position, it indicates we are not getting power to that axis. In this case, take another look at the connection of the wire to that axis and make sure the connectors are fully seated in both the Control Box and at the motor. Odds are though, the driver has gone bad. If you give us a call we will help you test this out and quickly get you a new driver if necessary.

Dealing with Tool Operation Problems: Qualitative Issues

The second kind of problem you may experience is more subtle. Your ShopBot is basically working -- but something is not working quite right. This problem can reveal itself in the tool having slight positioning problems or with cuts that are not of the quality you would like. Generally speaking, electronics and software usually work or they don't work. Thus positioning and quality problems are usually related to the mechanical function or 'tuning' of the tool. Listed below are areas of mechanical operation you should check (you might also check the maintenance section above).

Bits, bits, bits ... You'll probably get tired of hearing this, but bit quality and sharpness is incredibly important to successful cutting with your ShopBot. Your tool is capable of generating about 60 pounds of force as it moves the cutter. This is usually enough to plow through most material, or break a bit trying. However, you typically do not want to be cutting with this much force. The more force you put on the cutter, the more the cutter flexes and your ShopBot flexes. Such flex creates inaccuracies and increases the likelihood of uneven cutting and chatter. The best way to reduce this force is to use sharp cutters and to determine optimal cutting speeds and number of passes for the cutter. Plan to renew cutters frequently for quality cutting, and remember that with a CNC tool it is easy to accumulate mileage on a bit.

Ride. The way in which the carriages of your tool ride the track are critical for good cuts. Pull the X and Y motors away form the rack and evaluate the motion of the carriages. The carriages should move freely on the track. Look at the bearings, are they all exactly in their grooves. Is the pressure of the hold-down bearings on the Y axis appropriately tight to resist racking (all bearings should be engaged, you should be only just able to rotate them with your fingers). For the Z axis, is the carriage tight and resistant to flex and wobble in each direction.

Motor Engagement. Move the motors back into position with the pinion gear fully engaging the rack. Grease the rack. Now start the software and activate the motors. With the motors on but stopped, try to wiggle the carriage. Are all elements of the drive train tight and locked in position? Be particularly attentive to a set-screw being potentially loose on a shaft and allowing the pinion to move on the shaft. If you have a PRS with gearbox motors there is a small amount of backlash in the gearbox, but this should be a fraction of a degree.

Positioning. PRSalpha motors are closed-loop so positioning problems should be minimal. The information in this section is primarily for PRSstandards. If your tool seems to lose its location by a small amount, try to determine when and where this is happening. If happens only when you are cutting material, then it is probably a problem related to the forces of cutting at Move Speed. If it also happens when you are just air cutting, it probably has to do with positioning and tool action at Jog Speed. In either case, it is important to determine whether you are experiencing some sort of mechanical flex during the cutting, or whether the stepper motor is losing steps and thus losing track of its location. Go to the 0 point for each axis and mark or measure the carriage location on the track. Then run through the procedure that generates the location problem. Send the tool back to the 0 point and look at your marks. Assuming that you reproduced your cutting error, has the tool returned to its exact 0 location or is it now off position? If you are back in the right zero location, then the cutting errors must be a result of some sort of flex in the tool (assuming that the tool

 Copyright 2008 ShopBot Tools, Inc

moves the right distance when air-cutting … see above). If you are not back at the 0 location, then you tool has lost steps and lost position. If you lost steps and the problem happened while you were cutting material and not when you air cut, then you do not have enough power at the speed you are cutting to support that machining operation. The cutting may be too aggressive. (We are assuming that you previously eliminated the possibility of binding by confirming that your carriages are moving smoothly and the pinion position and pressure correct.) If you are cutting at a 1" per second or slower, your tool should have considerable force (say 60-75lb) ... if not, give us a call to help figure out what's going on. If you missed steps at Jog Speed, try giving your tool more margin by reducing the Jog Speed.

Position can also be lost as a result of electrical noise (voltage spikes) inadvertently triggering stepper driver electronics. The effects of electrical noise can usually be observed as occasional erratic stepping during air-cutting of a part. Electrical noise is typically only seen in shops with considerable heavy electrical equipment and is usually eliminated by putting the computer and ShopBot Control Box on a separate circuit and running them through a UPS (Un-interruptible Power Supply) or line conditioner.

TIP: *The Psychology of Troubleshooting.* Problems with your tool never occur when you have lots of time to enjoy solving them ... When they happen, you will usually be stressed out by other things going wrong at the same time and you will, no doubt, be in the middle of a production crisis. It's easy to say 'be cool' but we know how hard that is to do when the 'gol darned thing just won't work ...!'

Practically speaking, to fix things efficiently, what we really need to figure out is whether we need to send you a part, or whether something just needs to be changed or tweaked on your tool or with your software. That's it. So kick the tool or kick us ... then let's get down to effectively figuring things out. What we want to avoid is going off in the wrong diagnostic direction because we're not thinking clearly or understanding clearly what kind of a problem it really is.

Remember that when you e-mail, FAX, or call us that we are coming at the problem cold. We need to quickly know as much about the situation as possible to be most useful. We have found that first sending us an e-mail is the best way to handle problem. It can be especially efficient since you can lay out the problem and attach any relevant files as well as the PROBLEM.LOG file from your ShopBot folder; and we can look up the details of your tool. We continuously monitor e-mail during the workday, and regularly check it in the evenings and on the weekends. Go to the support section of our web site, or address the e-mail to: support@ShopBotTools.com

Getting Help

If you are having an assembly, mechanical, or electronic problem with your ShopBot, you may also want to check the ShopBot Forum on our web site for ideas on handling problems: http://www.shopbottools.com

If you haven't already tried it, as an initial step, try restarting your ShopBot Control Box, your computer, and the ShopBot software. If you cannot get the software to start, try deleting the configuration file (SHOPBOT.INI; a new configuration file will be generated).

Whenever you communicate with us about a problem, it is useful to:

1. Provide all the detail that you can about the exact steps that led up to a particular problem;

 Copyright 2008 ShopBot Tools, Inc

2. E-mail us a sample of any file that may be related to the problem or illustrate the problem; and

3. Send along the file named PROBLEM.LOG that you will find in your ShopBot Folder/Directory (this file allows us to re-create your exact set-up).

ShopBot Support is available:
BY email: support@shopbottools.com
BY fax: 919 680-4900
BY phone: 919 680-4800

Copyright 2008 ShopBot Tools, Inc

ShopBot FAX FORM

FAX Tech Support Request to us at: 919 680-4900

Your Name _____ Date _____ Time _____
Company (or other name that we may have your tool listed by) _____
E-mail _____

Your FAX _____
Work Phone (number we should call first) _____
Alternate Phone (number we should call second) _____

How would you prefer us to contact you? _____
Best time if phone … _____

Type of ShopBot (Size etc…) _____
Relevant accessories or modifications (how may channels, anything unusual ?)
_____
Version of Software _____

Brief description or category of problem (software, mechanical, electrical, problem with specific file or operation?): _____

Details of the problem (how did the problem come up, is this an initial start-up problem or has the tool been running normally and something has happened) :
_____
_____
_____

Have you already discussed problem with anyone at Shopbot Tools? _____

***One of the most helpful things you can do to speed the resolution of your problem is to send us (by e-mail or FAX) a copy of the file PROBLEM.LOG which you will find in your ShopBot 3 folder, along with any Part File that you are having a problem with. It helps us if you put your last name and date/time in the Subject field of email messages.

After hours, we are likely to get your request sooner if you e-mail the following information to us or fill out the Tech Support Form on our web site. We try to check e-mail every evening and periodically on weekends. During the day, even if your problem is best resolved with a phone call, it is more efficient for you to e-mail or FAX us the basic info here so that we can have the most appropriate support person contact you … already prepared for your specific issue.

Copyright 2008 ShopBot Tools, Inc

Technical Information

Specifications

Capacity and Rates:

- Standard tool accommodates sheet material up to 4' x 8' with cutting/machining depth (Z-axis) of 6 inches. Table size approximately 6' by 10'. Versions with work areas as small as 24" x 32" and as large as 5' by 12' are also standard.
- Step resolution is .0005" or better depending on gear ratio.
- Positioning accuracy is +/-.005" or better (no load).
- Positioning repeatability +/-.003" or better (no load).
- Cutting accuracy is approximately +/-.015" for heavy cutting applications. Reproduction accuracy depends on precision of setup and alignment, but in the same range or better.
- Simultaneous linear interpolation in 3 dimensions - 3D movement.
- X, Y cutting movement speed is variable with good cutting power up to approximately 240"(standard)/600"(alpha) per minute.
- X, Y rapid positioning speeds ('jogging') up to 600"(standard)/1800"(alpha) per minute.
- Z axis move speed up to 360" per minute.
- Linear force 60-75(standard)/150-200(alpha) lb. at 60" per minute, depending on gearing.

Tool Carrying Features:

- Tool Carrier is a universal aluminum extrusion that can be fitted with a standard router (e.g., Porter Cable 3.25 HP), industrial spindle (5hp or less), trim router, drill or other power tool. Mounting plate is provided for spindle equipped machines.
- A second Z axis or pneumatic drill can also be added to the tool bay.
- Easy screw-down, clamp-down, or tape-down of materials to be cut when using a typical plywood or particle-board tabletop. User can also add vacuum hold down.

Construction:

ShopBot PRS Tool is user-assembled from modular components. Major structure of the ShopBot PRS96 is aluminum and formed steel. Electronic components include four stepper motors, a Control Box with circuit boards, stepper motor drivers, and an input/output interface to the computer. All mechanical and electronic parts are cut and pre-drilled and all necessary hardware is provided. User supplies/purchases router and computer, and provides materials for work bed.

Control, Computer Interface and Software:

- Controlled via the USB port of a PC type computer. The computer is supplied by the user (an 800 MHz processor or higher is required with 512M RAM). Will work with Windows XP (XP Pro recommended) or Vista Business or Ultimate (64bit not supported)
- Limit/Stop detectors are provided for X and Y axes and permit automatic calibration or re-calibration, or specialized positioning operations.
- Eight output switches (5 volt; TTL level) that can be activated from software to switch relays and external devices (the first switch is dedicated to controlling router

 Copyright 2008 ShopBot Tools, Inc

or spindle on a PRSalpha tool, a second switch is automatically activated when tool is in motion).
- Eight inputs for sensing switch closure from software. Tow of these are optionally designated for 'limit' inputs and one for a Stop switch input.
- Optional 4th or 5th Axis control channels for additional stepper motor devices (e.g. a second Z axis or a rotary axis for turning parts; etc...)
- ShopBot Control Software reads .sbp (OpenSBP) Part Files as well as standard G-code and includes a .dxf converter which allows conversion of files from CAD programs such as TurboCAD, DesignCAD, AutoCAD, Corel Draw, and SignLab. Also includes HPGL, and bitmap converters.
- A 'Template' file mode is available that allows easy cutting of 2D outlines that require multiple passes and provides an additional capability of automatically putting "tabs" in the cutout to hold parts in position.
- ShopBot Control Software has a 'Preview' Mode for viewing Part Files before they are cut, or when the computer is not attached to a tool.
- Keyboard commands are all easily learned, 2-keystroke actions with prompts and HELP available, including the full Command Reference Manual.
- Full positioning control of the tool from the keyboard using the arrow keys.
- The same 2-keystroke commands are used in Part File. The Part File Programming language (OpenSBP) includes additional programming and customization capabilities.
- Editor and Calculator accessible through icons.

These specifications are subject to change.

ShopBot Computer Requirements

ShopBot Control Software Version 3 requires a PC running Windows XP (XP Pro recommended) or Vista Business or Ultimate (64bit not supported). For designing and responsive preview graphics, you will be happiest with an 800 MHz or faster computer and 512M RAM.

Graphic displays of 800 x 600, 1024 x 768, or 1400 x 1050 resolution all work. To reduce eye strain, we recommend the 1400 x 1050 for work in the office, and 1024 X 768 for work in the shop.

The Computer must have a USB (USB2 capable preferred) port that can be dedicated to running ShopBot. A hub and USB cable is included for connection between the computer and control box.

If you are running the ShopBot PRS from a laptop computer, always have the computer plugged in. Do not attempt to run the ShopBot PRS under battery alone.

Electrical Noise generated by other shop equipment such as compressors and saws can sometimes interfere with the operation of your ShopBot. This is because a brief power spike can disrupt the serial communication between your computer and control box. An uninterruptible power supply (UPS) for the computer and Control Box will usually eliminate such problems if you experience them. It is also best to place your ShopBot Control Box and computer on a separate circuit from the router and dust-collector and other shop equipment.

Copyright 2008 ShopBot Tools, Inc

ShopBot Philosophy

ShopBot is a workshop robot -- a Personal Robotic Tool. ShopBot's development has been guided by the principle that technology can contribute to creativity in exciting ways -- without excessive costs or technological dependencies. In industrial settings robots increase productivity through automation of manufacturing. For the small professional shop or for the individual craftsperson not only can ShopBot automate repetitive production tasks, but the same technology can enhance capabilities and provide new freedom of expression. At ShopBot we are dedicated to giving you access to these new tools for productivity and creativity.

By 'accessible' we mean affordable ... but we also intend our tools to be simple and robust, easily maintained, and non-obsolescent. Most ShopBots come as 4 pre-assembled components. While completing the assembly of your ShopBot you will become familiar with its parts, how it works, and what will be involved in keeping it adjusted. The control system is designed to be quickly learned, though it has advanced functions available when you are ready for them. We do continue to improve and develop ShopBots, yet we work to make all improvements available for existing tools, although this is not always possible with all upgrades. Where an upgrade is significant it will be available for purchase from us at a reasonable price.

We are committed to supporting ShopBot with help and information to our customers. We will encourage designers to develop plans and projects for the tool and encourage developers to create special-purpose software for fully utilizing ShopBot's capabilities in small business and production settings. If you have questions about your tool, suggestions for software or projects you would like to see converted to cutting files -- call us at 919 680-4800 or visit our web site at www.ShopBotTools.com.

Copyright 2008 ShopBot Tools, Inc